XBee™ ZNet 2.5/XBee-PRO™ ZNet 2.5 OEM RF Modules

XBee ZNet 2.5/XBee PRO Znet 2.5 OEM RF Modules

ZigBee™ Networks

RF Module Operation

RF Module Configuration

Appendices

Product Manual v1.x.4x - *ZigBee* Protocol

For OEM RF Module Part Numbers: XB24-BxIT-00x

ZigBee OEM RF Modules by Digi International

Firmware Versions:
- 1.0xx - Coordinator, Transparent Operation
- 1.1xx - Coordinator, API Operation
- 1.2xx - Router, End Device, Transparent Operation
- 1.3xx - Router, End Device, API Operation

Digi International Inc.
11001 Bren Road East
Minnetonka, MN 55343
877 912-3444 or 952 912-3444
http://www.digi.com

90000866_C
2/11/2008

© 2008 Digi International, Inc. All rights reserved

No part of the contents of this manual may be transmitted or reproduced in any form or by any means without the written permission of Digi International, Inc.

ZigBee® is a registered trademark of the ZigBee Alliance.

XBee™ ZNet 2.5/XBee™ ZNet 2.5 PRO is a trademark of Digi International, Inc.

Technical Support: Phone: (801) 765-9885

Live Chat: www.digi.com

E-mail: support@digi.com

Contents

1. Overview 4

1.1. Key Features 5
1.1.1. Worldwide Acceptance 5
1.2. Specifications 5
1.3. Mechanical Drawings 7
1.4. Mounting Considerations 8
1.5. Pin Signals 9
1.6. Electrical Characteristics 10

2. RF Module Operation 11

2.1. Serial Communications 11
2.1.1. UART Data Flow 11
2.1.2. Serial Buffers 11
2.1.3. Serial Flow Control 12
2.1.4. Serial Interface Protocols 13
2.2. Modes of Operation 14
2.2.1. Idle Mode 14
2.2.2. Transmit Mode 14
2.2.3. Receive Mode 15
2.2.4. Command Mode 15
2.2.5. Sleep Mode 16

3. ZigBee Networks 17

3.1. ZigBee Network Formation 17
3.2. ZigBee PANs 17
3.2.1. Starting a PAN 17
3.3. Joining a PAN 18
3.3.1. Allowing Joining 19
3.3.2. Security 20
3.4. ZigBee Network Communications 21
3.4.1. ZigBee Device Addressing 21
3.4.2. ZigBee Application-layer Addressing 21
3.4.3. Data Transmission and Routing 22

4. XBee ZNet 2.5 Networks 27

4.1. XBee ZNet 2.5 Network Formation 27
4.1.1. Starting an XBee ZNet 2.5 Coordinator 27
4.1.2. Joining an XBee ZNet 2.5 Router or End Device to a PAN 28
4.1.3. Verifying Network Selection 29
4.1.4. Secure Networks 29
4.1.5. Open and Closed Networks 29
4.1.6. Resetting Network Parameters 30
4.2. XBee ZNet 2.5 Addressing 31
4.2.1. Device Addressing 31
4.2.2. Application-layer Addressing 33
4.3. Sleeping End Devices 34
4.3.1. End Device Operation 34
4.3.2. Parent Operation 35
4.3.3. End Device Behavior 37
4.3.4. Parent Behavior 37
4.3.5. End Device Sleep Configuration 37
4.4. Remote Configuration Commands 41
4.4.1. Sending a Remote Command 41
4.4.2. Applying Changes on Remote 41
4.4.3. Remote Command Responses 41
4.5. IO Line Monitoring 41
4.5.1. IO Samples 43
4.5.2. Queried Sampling 44
4.5.3. Periodic IO Sampling 44
4.5.4. Digital IO Change Detection 44
4.5.5. Voltage Supply Monitoring 45
4.6. I/O Line Configuration 45

5. Network Commissioning and Diagnostics 48

5.1. Device Discovery 48
5.2. Device Configuration 48
5.3. Device Placement 48
5.3.1. Link Testing 48
5.3.2. RSSI Indicators 49
5.4. Commissioning Pushbutton and Associate LED 49
5.4.1. Commissioning Pushbutton 50
5.4.2. Associate LED 51

6. API Operation 54

6.0.1. API Frame Specifications 54
6.0.2. API Examples 56
6.0.3. Supporting the API 57
6.0.4. API Frames 57

7. XBee ZNet 2.5 Command Reference Tables 70

8. Manufacturing Support 78

8.1. Interoperability with other EM250 Devices 78
8.1.1. Crosstalk Issue 78
8.2. Customizing XBee Default Parameters 78
8.3. XBee EM250 Pin Mappings 78
8.4. XBee ZNet 2.5 Custom Bootloader 79
8.5. Programming XBee ZNet 2.5 Modules 80
8.6. Developing Custom Firmware 81
8.7. Design Considerations for Digi Drop-in

© 2008 Digi Internaitonal, Inc.

Contents

Networking 81

Appendix I:Definitions 82

Appendix J: Agency Certifications 84

Appendix K:Migrating from the 802.15.4 Protocol 89

Appendix L:Additional Information 90

1. Overview

The XBee/XBee-PRO ZNet 2.5 OEM (formerly known as Series 2 and Series 2 PRO) RF Modules were engineered to operate within the ZigBee protocol and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between remote devices.

The modules operate within the ISM 2.4 GHz frequency band and are compatible with the following:

- XBee RS-232 Adapter
- XBee RS-232 PH (Power Harvester) Adapter
- XBee RS-485 Adapter
- XBee Analog I/O Adapter
- XBee Digital I/O Adapter
- XBee Sensor Adapter
- XBee USB Adapter
- XStick
- ConnectPort X Gateways
- XBee Wall Router.

© 2008 Digi International, Inc.

1.1. Key Features

High Performance, Low Cost

- Indoor/Urban: up to 300' (100 m)
- Outdoor line-of-sight: up to 1 mile (1.6 km)
- Transmit Power Output: 100 mW (20 dBm) EIRP
- Receiver Sensitivity: -102 dBm

RF Data Rate: 250,000 bps

Advanced Networking & Security

Retries and Acknowledgements

DSSS (Direct Sequence Spread Spectrum)

Each direct sequence channel has over 65,000 unique network addresses available

Point-to-point, point-to-multipoint and peer-to-peer topologies supported

Self-routing, self-healing and fault-tolerant mesh networking

Low Power

XBee PRO ZNet 2.5

- TX Current: 295 mA (@3.3 V)
- RX Current: 45 mA (@3.3 V)
- Power-down Current: < 1 µA @ 25°C

Easy-to-Use

No configuration necessary for out-of box RF communications

AT and API Command Modes for configuring module parameters

Small form factor

Extensive command set

Free X-CTU Software (Testing and configuration software)

Free & Unlimited Technical Support

1.1.1. Worldwide Acceptance

FCC Approval (USA) Refer to Appendix A [p50] for FCC Requirements. Systems that contain XBee /XBee-PRO ZNet 2.5 RF Modules inherit Digi Certifications.

ISM (Industrial, Scientific & Medical) **2.4 GHz frequency band**

Manufactured under **ISO 9001:2000** registered standards

XBee /XBee-PRO ZNet 2.5 RF Modules are optimized for use in **US**, **Canada**, **Australia, Israel and Europe** (contact MaxStream for complete list of agency approvals).

1.2. Specifications

Table 1-01. Specifications of the XBee /XBee-PRO ZNet 2.5 OEM RF Module

Specification	XBee ZNet 2.5	XBee PRO ZNet 2.5
Performance		
Indoor/Urban Range	up to 133 ft. (40 m)	up to 300 ft. (100 m)
Outdoor RF line-of-sight Range	up to 400 ft. (120 m)	up to 1 mile (1.6 km)
Transmit Power Output	2mW (+3dBm), boost mode enabled 1.25mW (+1dBm), boost mode disabled	63mW (+18 dBm) 10mW (+10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 - 230400 bps (non-standard baud rates also supported)	1200 - 230400 bps (non-standard baud rates also supported)
Receiver Sensitivity	-96 dBm, boost mode enabled -95 dBm, boost mode disabled	-102 dBm

© 2008 Digi International, Inc.

Table 1-01. Specifications of the XBee /XBee-PRO ZNet 2.5 OEM RF Module

Specification	XBee ZNet 2.5	XBee PRO ZNet 2.5
Power Requirements		
Supply Voltage	2.1 - 3.6 V	3.0 - 3.4 V
Operating Current (Transmit, max output power)	40mA (@ 3.3 V, boost mode enabled) 35mA (@ 3.3 V, boost mode disabled)	295mA (@3.3 V)
Operating Current (Receive))	40mA (@ 3.3 V, boost mode enabled) 38mA (@ 3.3 V, boost mode disabled)	45 mA (@3.3 V)
Idle Current (Receiver off)	15mA	15mA
Power-down Current	< 1 uA @ 25°C	< 1 uA @ 25°C
General		
Operating Frequency Band	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960 x 1.297 (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip, RPSMA, or U.FL Connector*	Integrated Whip, Chip, RPSMA, or U.FL Connector*
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh
Number of Channels	16 Direct Sequence Channels	13 Direct Sequence Channels
Addressing Options	PAN ID and Addresses, Cluster IDs and Endpoints (optional)	PAN ID and Addresses, Cluster IDs and Endpoints (optional)
Agency Approvals		
United States (FCC Part 15.247)	OUR-XBEE2	MCQ-XBEEPRO2
Industry Canada (IC)	4214A-XBEE2	1846A-XBEEPRO2
Europe (CE)	ETSI	ETSI
RoHS	Compliant	Compliant

© 2008 Digi International, Inc.

1.3. Mechanical Drawings

Figure 1-01. Mechanical drawings of the XBee /XBee-PRO ZNet 2.5 OEM RF Modules (antenna options not shown)

.

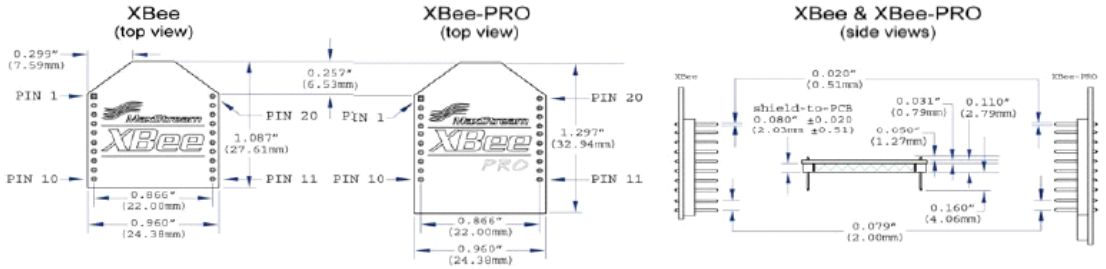
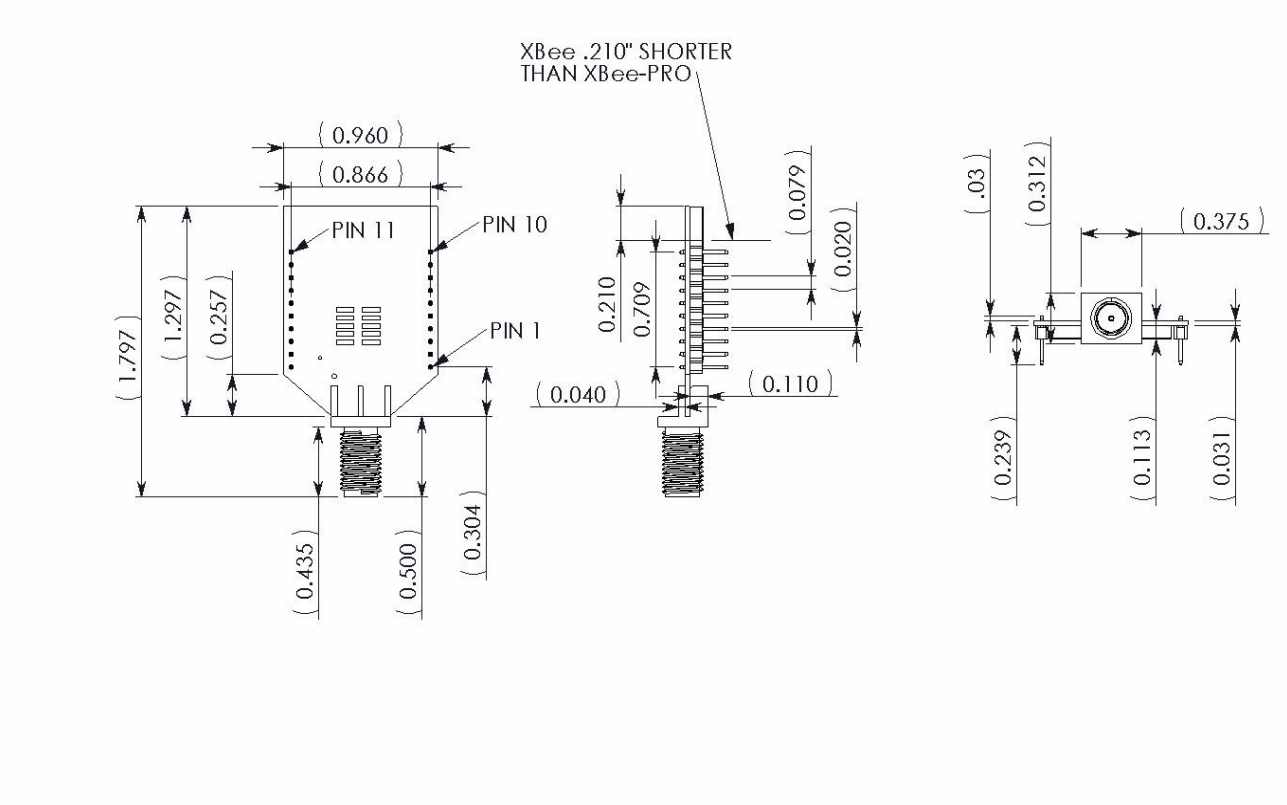


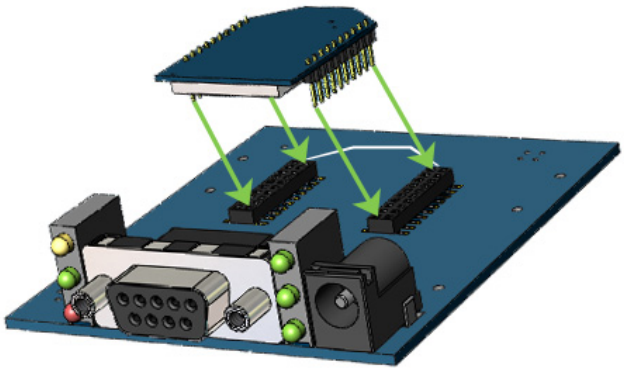
Figure 1-02. Mechanical Drawings for the RPSMA Variant



1.4. Mounting Considerations

The XBee modules were designed to mount into a receptacle (socket) and therefore does not require any soldering when mounting it to a board. The XBee PRO ZNet 2.5 Development Kits contain RS-232 and USB interface boards which use two 20-pin receptacles to receive modules.

Figure 1-03. XBee PRO ZNet 2.5 Module Mounting to an RS-232 Interface Board.



The receptacles used on Digi development boards are manufactured by Century Interconnect. Several other manufacturers provide comparable mounting solutions; however, Digi currently uses the following receptacles:

- Through-hole single-row receptacles -
  Samtec P/N: MMS-110-01-L-SV (or equivalent)
- Surface-mount double-row receptacles -
  Century Interconnect P/N: CPRMSL20-D-0-1 (or equivalent)
- Surface-mount single-row receptacles -
  Samtec P/N: SMM-110-02-SM-S

Digi also recommends printing an outline of the module on the board to indicate the orientation the module should be mounted.

1.5. Pin Signals

Figure 1-04. XBee /XBee-PRO ZNet 2.5 RF Module Pin Number
(top sides shown - shields on bottom)

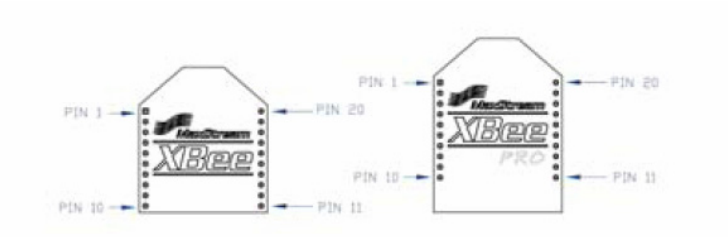


Table 1-02. Pin Assignments for the XBee PRO ZNet 2.5 Modules
(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / **CONFIG**	Input	UART Data In
4	DIO12	Either	Digital I/O 12
5	**RESET**	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI / DIO10	Either	PWM Output 0 / RX Signal Strength Indicator / Digital IO
7	PWM / DIO11	Either	Digital I/O 11
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ/ DIO8	Either	Pin Sleep Control Line or Digital IO 8
10	GND	-	Ground
11	DIO4	Either	Digital I/O 4
12	**CTS** / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / **SLEEP** / **DIO9**	Output	Module Status Indicator or Digital I/O 9
14	[reserved]	-	Do not connect
15	Associate / DIO5	Either	Associated Indicator, Digital I/O 5
16	**RTS** / DIO6	Either	Request-to-Send Flow Control, Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0 / Commissioning Button	Either	Analog Input 0, Digital IO 0, or Commissioning Button

Design Notes:

- Minimum connections: VCC, GND, DOUT & DIN
- Minimum connections to support serial firmware upgrades: VCC, GND, DIN, DOUT, RTS & DTR
- Signal Direction is specified with respect to the module
- Module includes a 30k Ohm resistor attached to **RESET**
- Several of the input pull-ups can be configured using the PR command
- Unused pins should be left disconnected
- Pin 20 can be connected to a push button (pin grounded when closed) to support the commissioning push button functionality. See "Commissioning Pushbutton and Associate LED" for details.

© 2008 Digi International, Inc.

1.6. Electrical Characteristics

Table 1-03. DC Characteristics of the XBee PRO ZNet 2.5 (VCC = 3.0 - 3.4 VDC).

Symbol	Parameter	Condition	Min	Typical	Max	Units
VIL	Input Low Voltage	All Digital Inputs	-	-	0.2 * VCC	V
V_{IH}	Input High Voltage	All Digital Inputs	0.8 * VCC	-	-	V
V_{OL}	Output Low Voltage	I_{OL} = 2 mA, VCC >= 2.7 V	-	-	0.18*VCC	V
V_{OH}	Output High Voltage	I_{OH} = -2 mA, VCC >= 2.7 V	0.82*VCC	-	-	V
II_{IN}	Input Leakage Current	V_{IN} = VCC or GND, all inputs, per pin	-	-	0.5uA	uA

© 2008 Digi International, Inc.

2. RF Module Operation

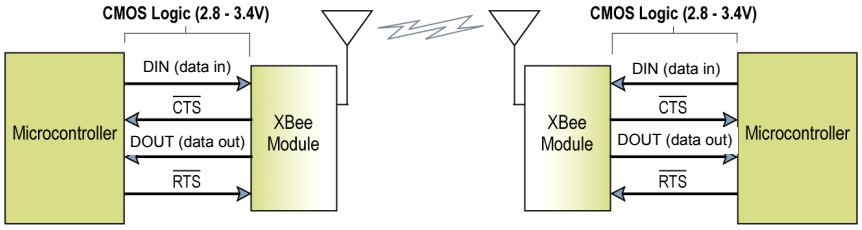
2.1. Serial Communications

The XBee ZNet 2.5 OEM RF Modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device (For example: Through a Digi proprietary RS-232 or USB interface board).

2.1.1. UART Data Flow

Devices that have a UART interface can connect directly to the pins of the RF module as shown in the figure below.

Figure 2-01. System Data Flow Diagram in a UART-interfaced environment
(Low-asserted signals distinguished with horizontal line over signal name.)

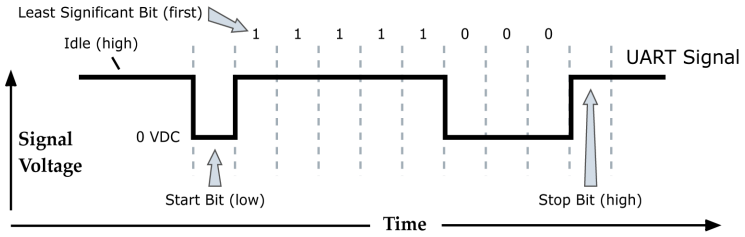


Serial Data

Data enters the module UART through the DIN (pin 3) as an asynchronous serial signal. The signal should idle high when no data is being transmitted.

Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following figure illustrates the serial bit pattern of data passing through the module.

Figure 2-02. UART data packet 0x1F (decimal number "31") as transmitted through the RF module
Example Data Format is 8-N-1 (bits - parity - # of stop bits)

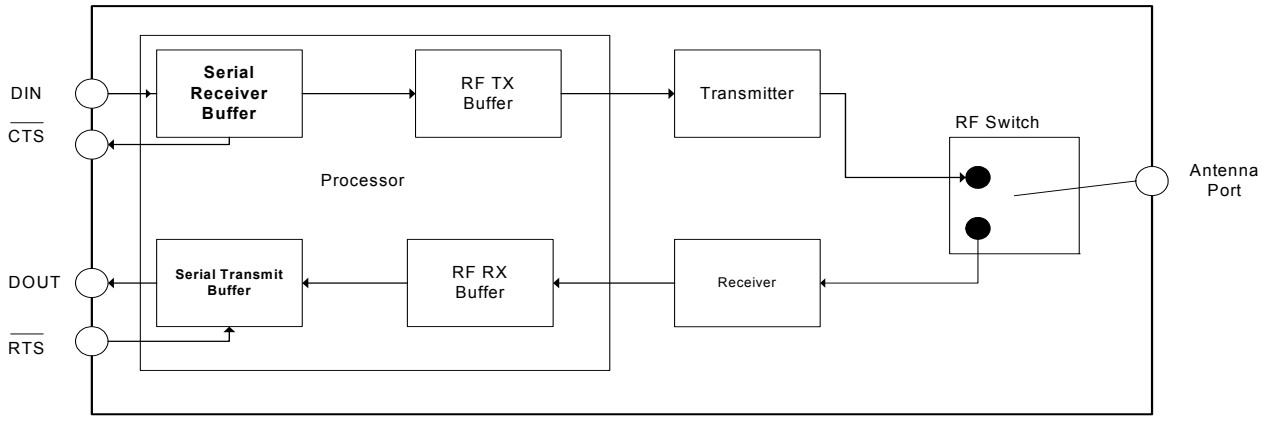


The module UART performs tasks, such as timing and parity checking, that are needed for data communications. Serial communications depend on the two UARTs to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).

2.1.2. Serial Buffers

The XBee ZNet 2.5 modules maintain small buffers to collect received serial and RF data, which is illustrated in the figure below. The serial receive buffer collects incoming serial characters and holds them until they can be processed. The serial transmit buffer collects data that is received via the RF link that will be transmitted out the UART.

Figure 2-03. Internal Data Flow Diagram



Serial Receive Buffer

When serial data enters the RF module through the DIN Pin (pin 3), the data is stored in the serial receive buffer until it can be processed. Under certain conditions, the module may not be able to process data in the serial receive buffer immediately. If large amounts of serial data are sent to the module, $\overline{CTS}$ flow control may be required to avoid overflowing the serial receive buffer.

Cases in which the serial receive buffer may become full and possibly overflow:

1.If the module is receiving a continuous stream of RF data, the data in the serial receive buffer will not be transmitted until the module is no longer receiving RF data.

2.If the module is transmitting an RF data packet, the module may need to discover the destination address or establish a route to the destination. After transmitting the data, the module may need to retransmit the data if an acknowledgment is not received, or if the transmission is a broadcast. These issues could delay the processing of data in the serial receive buffer.

Serial Transmit Buffer

When RF data is received, the data is moved into the serial transmit buffer and sent out the UART. If the serial transmit buffer becomes full enough such that all data in a received RF packet won't fit in the serial transmit buffer, the entire RF data packet is dropped.

Cases in which the serial transmit buffer may become full resulting in dropped RF packets

1. If the RF data rate is set higher than the interface data rate of the module, the module could receive data faster than it can send the data to the host.

2. If the host does not allow the module to transmit data out from the serial transmit buffer because of being held off by hardware flow control.

2.1.3. Serial Flow Control

The $\overline{RTS}$ and $\overline{CTS}$ module pins can be used to provide $\overline{RTS}$ and/or $\overline{CTS}$ flow control. $\overline{CTS}$ flow control provides an indication to the host to stop sending serial data to the module. RTS flow control allows the host to signal the module to not send data in the serial transmit buffer out the uart. $\overline{RTS}$ and $\overline{CTS}$ flow control are enabled using the D6 and D7 commands.

$\overline{CTS}$ Flow Control

If $\overline{CTS}$ flow control is enabled (D7 command), when the serial receive buffer is 17 bytes away from being full, the module de-asserts $\overline{CTS}$ (sets it high) to signal to the host device to stop sending serial data. $\overline{CTS}$ is re-asserted after the serial receive buffer has 34 bytes of space.

$\overline{\text{RTS}}$ Flow Control

If RTS flow control is enabled (D6 command), data in the serial transmit buffer will not be sent out the DOUT pin as long as $\overline{\text{RTS}}$ is de-asserted (set high). The host device should not de-assert $\overline{\text{RTS}}$ for long periods of time to avoid filling the serial transmit buffer. If an RF data packet is received, and the serial transmit buffer does not have enough space for all of the data bytes, the entire RF data packet will be discarded.

2.1.4. Serial Interface Protocols

The XBee modules support both transparent and API (Application Programming Interface) serial interfaces.

Transparent Operation

When operating in transparent mode, the modules act as a serial line replacement. All UART data received through the DIN pin is queued up for RF transmission. When RF data is received, the data is sent out through the DOUT pin. The module configuration parameters are configured using the AT command mode interface.

Data is buffered in the serial receive buffer until one of the following causes the data to be packetized and transmitted:

1. No serial characters are received for the amount of time determined by the RO (Packetization Timeout) parameter. If RO = 0, packetization begins when a character is received.

2. Maximum number of characters that will fit in an RF packet is received (72 bytes).

3. The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the serial receive buffer before the sequence is transmitted.

RF modules that contain the following firmware versions will support Transparent Mode: 1.0xx (coordinator) and 1.2xx (router/end device).

API Operation

API operation is an alternative to transparent operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module. When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module.

Transmit Data Frames (received through the DIN pin (pin 3)) include:

- RF Transmit Data Frame
- Command Frame (equivalent to AT commands)

Receive Data Frames (sent out the DOUT pin (pin 2)) include:

- RF-received data frame
- Command response
- Event notifications such as reset, associate, disassociate, etc.

The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and payload information instead of using command mode to modify addresses. The module will send data frames to the application containing status packets; as well as source, and payload information from received data packets.

The API operation option facilitates many operations such as the examples cited below:

-> Transmitting data to multiple destinations without entering Command Mode

-> Receive success/failure status of each transmitted RF packet

-> Identify the source address of each received packet

RF modules that contain the following firmware versions will support API operation: 1.1xx (coordinator) and 1.3xx (router/end device).

© 2008 Digi International, Inc.

2.2. Modes of Operation

2.2.1. Idle Mode

When not receiving or transmitting data, the RF module is in Idle Mode. During Idle Mode, the RF module is also checking for valid RF data. The module shifts into the other modes of operation under the following conditions:

- Transmit Mode (Serial data in the serial receive buffer is ready to be packetized)
- Receive Mode (Valid RF data is received through the antenna)
- Sleep Mode (End Devices only)
- Command Mode (Command Mode Sequence is issued)

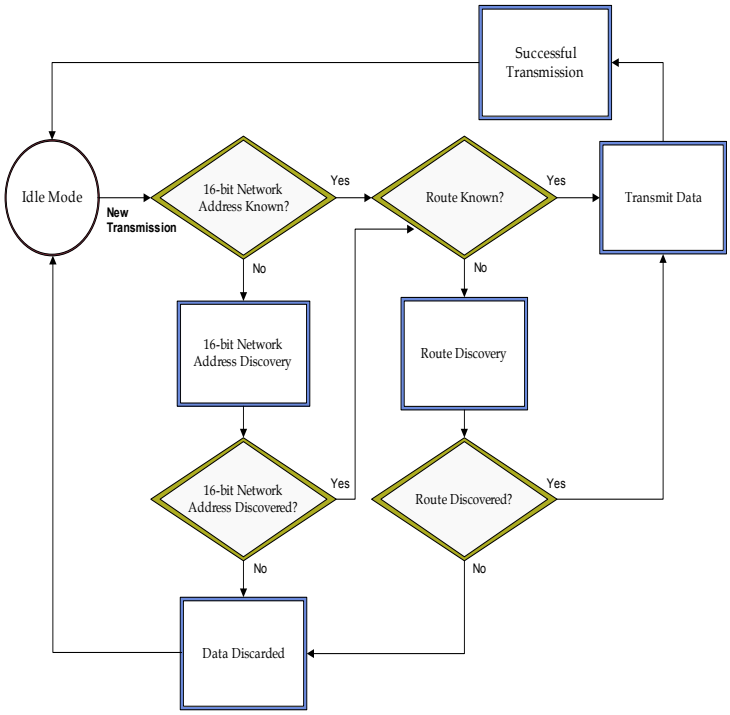
2.2.2. Transmit Mode

When serial data is received and is ready for packetization, the RF module will exit Idle Mode and attempt to transmit the data. The destination address determines which node(s) will receive the data.

Prior to transmitting the data, the module ensures that a 16-bit network address and route to the destination node have been established.

If the destination 16-bit network address is not known, network address discovery will take place. If a route is not known, route discovery will take place for the purpose of establishing a route to the destination node. If a module with a matching network address is not discovered, the packet is discarded. The data will be transmitted once a route is established. If route discovery fails to establish a route, the packet will be discarded.

Figure 2-04. Transmit Mode Sequence



When data is transmitted from one node to another, a network-level acknowledgement is transmitted back across the established route to the source node. This acknowledgement packet indicates to the source node that the data packet was received by the destination node. If a network acknowledgement is not received, the source node will re-transmit the data.

It is possible in rare circumstances for the destination to receive a data packet, but for the source to not receive the network acknowledgment. In this case, the source will retransmit the data, which could cause the destination to receive the same data packet multiple times. The XBee ZNet 2.5 modules do not filter out duplicate packets. The application should include provisions to address this potential issue

See Data Transmission and Routing in chapter 3 for more information.

2.2.3. Receive Mode

If a valid RF packet is received, the data is transferred to the serial transmit buffer.

2.2.4. Command Mode

To modify or read RF Module parameters, the module must first enter into Command Mode - a state in which incoming serial characters are interpreted as commands. Refer to the API Mode section in Chapter 7 for an alternate means of configuring modules.

AT Command Mode

To Enter AT Command Mode:

Send the 3-character command sequence "+++" and observe guard times before and after the command characters. [Refer to the "Default AT Command Mode Sequence" below.]

Default AT Command Mode Sequence (for transition to Command Mode):

- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]
- Input three plus characters ("+++") within one second [CC (Command Sequence Character) parameter = 0x2B.]
- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]

Once the AT command mode sequence has been issued, the module sends an "OK\r" out the DOUT pin. The "OK\r" characters can be delayed if the module has not finished transmitting received serial data.

When command mode has been entered, the command mode timer is started (CT command), and the module is able to receive AT commands on the DIN pin.

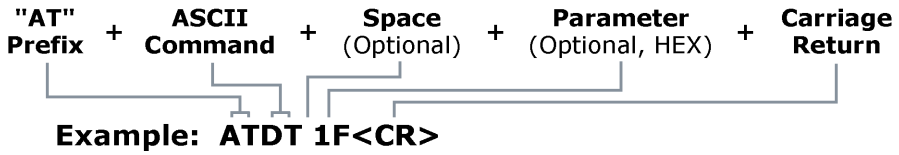
All of the parameter values in the sequence can be modified to reflect user preferences.

NOTE: Failure to enter AT Command Mode is most commonly due to baud rate mismatch. By default, the BD (Baud Rate) parameter = 3 (9600 bps).

To Send AT Commands:

Send AT commands and parameters using the syntax shown below.

Figure 2-05. Syntax for sending AT Commands



To read a parameter value stored in the RF module's register, omit the parameter field.

The preceding example would change the RF module Destination Address (Low) to "0x1F". To store the new value to non-volatile (long term) memory, subsequently send the WR (Write) command.

For modified parameter values to persist in the module's registry after a reset, changes must be saved to non-volatile memory using the WR (Write) Command. Otherwise, parameters are restored to previously saved values after the module is reset.

Command Response

When a command is sent to the module, the module will parse and execute the command. Upon successful execution of a command, the module returns an "OK" message. If execution of a command results in an error, the module returns an "ERROR" message.

Applying Command Changes

Any changes made to the configuration command registers through AT commands will not take effect until the changes are applied. For example, sending the BD command to change the baud rate will not change the actual baud rate until changes are applied. Changes can be applied in one of the following ways:

- The AC (Apply Changes) command is issued.
- AT command mode is exited.

To Exit AT Command Mode:

1. Send the ATCN (Exit Command Mode) command (followed by a carriage return).

   [OR]

2. If no valid AT Commands are received within the time specified by CT (Command Mode Timeout) Command, the RF module automatically returns to Idle Mode.

For an example of programming the RF module using AT Commands and descriptions of each configurable parameter, refer to the "Examples" and "XBee ZNet 2.5 Command Reference Tables" chapters.

2.2.5. Sleep Mode

Sleep modes allow the RF module to enter states of low power consumption when not in use. The XBee ZNet 2.5 OEM RF modules support both pin sleep (sleep mode entered on pin transition) and cyclic sleep (module sleeps for a fixed time). XBee ZNet 2.5 sleep modes are discussed in detail in section 5.3.

© 2008 Digi International, Inc.

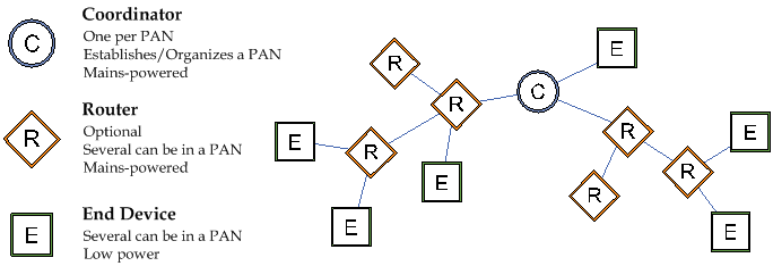
3. ZigBee Networks

3.1. ZigBee Network Formation

Zigbee networks are called personal area networks (PAN). Each network contains a 16-bit identifier called a PAN ID.

ZigBee defines three different device types – coordinator, router, and end device. An example of such a network is shown below.

Figure 3-01. Node Types / Sample of a Basic ZigBee Network Topology



Coordinator – Responsible for selecting the channel and PAN ID. The coordinator starts a new PAN. Once it has started a PAN, the coordinator can allow routers and end devices to join the PAN. The coordinator can transmit and receive RF data transmissions, and it can assist in routing data through the mesh network. Coordinators are not intended to be battery-powered devices. Since the coordinator must be able to allow joins and/or route data, it should be mains powered.

Router – A router must join a ZigBee PAN before it can operate. After joining a PAN, the router can allow other routers and end devices to join the PAN. The router can also transmit and receive RF data transmissions, and it can route data packets through the network. Since routers can allow joins and participate in routing data, routers cannot sleep and should be mains powered.

End Device – An end device must join a ZigBee PAN, similar to a router. The end device, however, cannot allow other devices to join the PAN, nor can it assist in routing data through the network. An end device can transmit or receive RF data transmissions. End devices are intended to be battery powered devices. Since the end device may sleep, the router or coordinator that allows the end device to join must collect all data packets intended for the end device, and buffer them until the end device wakes and is able to receive them. The router or coordinator that allowed the end device to join and that manages RF data on behalf of the end device is known as the end device's parent. The end device is considered a child of its parent.

3.2. ZigBee PANs

ZigBee networks are formed when a coordinator first selects a channel and PAN ID. After the coordinator has started the PAN, routers and end devices may join the PAN. The PAN ID is selected by the coordinator when it starts the PAN. Routers and end devices become a part of the PAN (and inherit the coordinator's PAN ID) when they join a PAN.

ZigBee supports mesh routing in the network, allowing data packets to traverse multiple nodes (multiple "hops") in order to reach the destination node. This allows ZigBee nodes to be spread out over a large region, and still support communications amongst all devices in the network.

All devices in a ZigBee network receive a 16-bit address (network address) when they join a PAN. The 16-bit address of the coordinator is always 0.

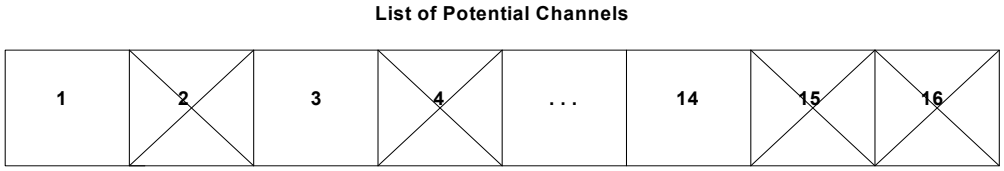
3.2.1. Starting a PAN

Since the coordinator is responsible for starting a ZigBee network, all ZigBee networks must have a coordinator present initially. To start a PAN, the coordinator performs a series of scans to

discover the level of RF activity on different channels (energy scan), and to discover any nearby operating PANs (PAN scan). Energy Scan

When a coordinator comes up for the first time, it performs an energy scan on multiple channels (frequencies) to detect energy levels on each channel. Channels with excessive detected energy levels are removed from its list of potential channels to start on.

Figure 3-02. Potential Channels

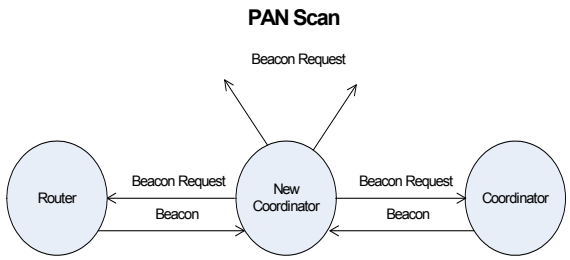


Performing an energy scan allows the coordinator to avoid starting on channels with high energy levels.

PAN Scan

When the energy scan completes, the coordinator scans the remaining quiet channels (found in the energy scan) for existing PANs. To do this, the coordinator sends a broadcast, one-hop beacon request. Any nearby coordinators and routers will respond to the beacon request by sending a beacon frame back to the coordinator. The beacon frame contains information about the PAN the sender is on, including the PAN identifier (PAN ID), and whether or not the device is allowing joining. (The PAN scan is more commonly called an active scan or a beacon scan.)

Figure 3-03. PAN Scans



A PAN Scan allows the coordinator to detect nearby PAN IDs to avoid duplicating existing PAN IDs

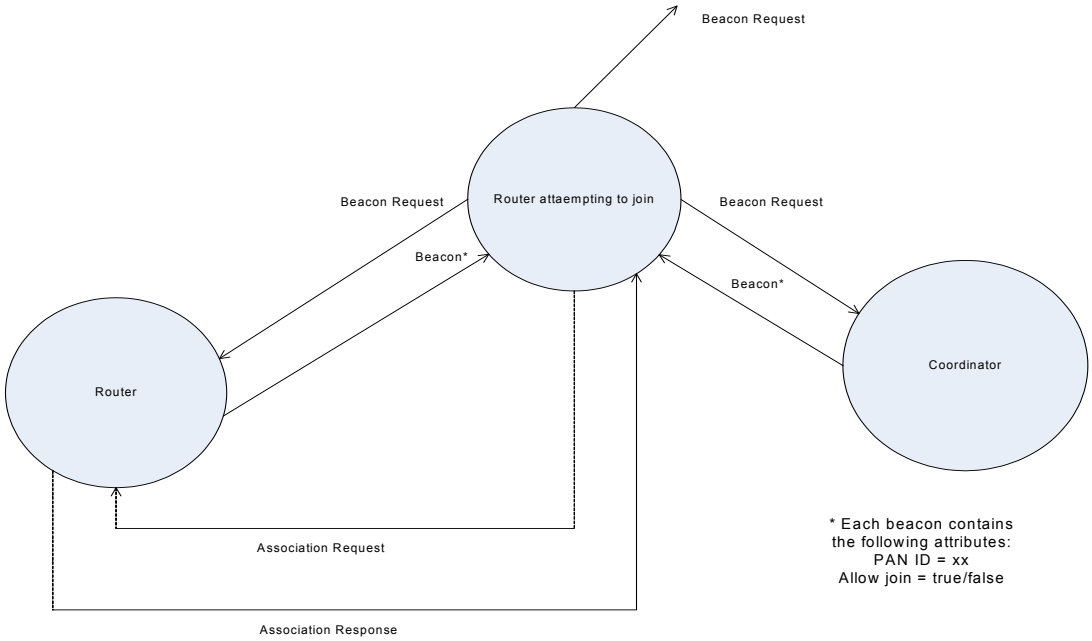
Once the coordinator has completed the energy and PAN scans, it parses all received beacons and attempts to start on an unused PAN ID and channel. When the coordinator starts a PAN, it can then allow routers and/or end devices to join the PAN. A coordinator retains the channel and PAN ID attributes through power cycle or reset events.

3.3. Joining a PAN

Router and end device types must discover and join a ZigBee PAN. To do this, they first issue a PAN scan, just like the coordinator does when it starts a PAN. From the PAN scan, the router or end device receives a list of beacons from nearby ZigBee devices. The router or end device parses this list to find a valid ZigBee network to join.

Routers and end devices can be configured to join any ZigBee PAN, or to only join a PAN with a certain PAN ID. However, they must always find a coordinator or router that is allowing joins.

Figure 3-04. Joining a PAN



A router or end device sends a beacon request to discover nearby Zigbee networks. If a device is found that is operating on a valid Zigbee network, that is allowing joins, the router or end device sends an association request to that device to attempt to join the network

Once a joining device (router or end device) discovers a device operating on a valid ZigBee network that is allowing joining, it attempts to join the PAN by sending an association request to that device.

3.3.1. Allowing Joining

The coordinator and all routers can allow new routers and end devices to join to them. Whether or not a particular coordinator or router will allow a new device to join depends upon two things:

- its permit-joining attribute (if joins are allowed)
- the number of end device children it already has

Permit-Joining Attribute

The coordinator and all routers have a permit-joining attribute. This attribute on a coordinator and any joined routers can be configured to always allow joins, allow joins for a short time, or to not allow any more joins. In order for a new device to join the network, this attribute must be set on a nearby device such that joins are enabled.

End Device Children

Since end devices rely on their parent router or coordinator to buffer incoming RF packets, the coordinator and each router can support a finite number of end device children. Once that number of end devices has joined a particular router or coordinator, the device can no longer allow end devices to join to it.

3.3.2. Security

If security is enabled, the coordinator will start up using a 128-bit AES encryption key. Only devices that have the same security key can communicate on the PAN. Routers and end devices that will join a secure PAN must obtain the correct security key. The security key can be obtained in one of two ways:

- pre-installation
- key is received over-the-air (in the clear) during joining

© 2008 Digi International, Inc.

3.4. ZigBee Network Communications

Zigbee supports device addressing and application layer addressing. Device addressing specifies the destination address of the device a packet is destined to. Application layer addressing indicates a particular application recipient, known as a Zigbee endpoint, along with a message type field called a Cluster ID.

3.4.1. ZigBee Device Addressing

The 802.15.4 protocol upon which the ZigBee protocol is built specifies two address types:

- 16-bit network addresses
- 64-bit Addresses

16-bit Network Addresses

A 16-bit network address is assigned to a node when the node joins a network. The network address is unique to each node in the network. However, network addresses are not static - it can change.

The following two conditions will cause a node to receive a new network address:

1. If an end device cannot communicate with its parent it may need to leave the network and rejoin to find a new parent.

2. If the device type changes from router to end device, or vice-versa, the device will leave the network and rejoin as the new device type.

ZigBee requires that data be sent to the 16-bit network address of the destination device. This requires that the 16-bit address be discovered before transmitting data. See 3.2.3 Network Address Discovery for more information.

64-bit Addresses

Each node contains a unique 64-bit address. The 64-bit address uniquely identifies a node and is permanent.

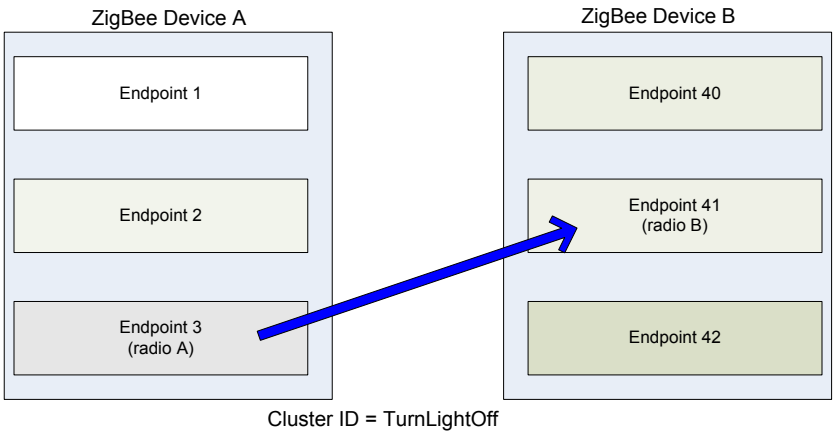
3.4.2. ZigBee Application-layer Addressing

The ZigBee application layers define endpoints and cluster identifiers (cluster IDs) that are used to address individual services or applications on a device. An endpoint is a distinct task or application that runs on a ZigBee device, similar to a TCP port. Each ZigBee device may support one or more endpoints. Cluster IDs define a particular function or action on a device. Cluster IDs in the ZigBee home controls lighting profile, for example, would include actions such as “TurnLightOn”, “TurnLightOff”, “DimLight”, etc.

Suppose a single radio controls a light dimmer and one or more light switches. The dimmer and switches could be assigned to different endpoint values. To send a message to the dimmer, a remote radio would transmit a message to the dimmer endpoint on the radio. In this example, the radio might support cluster IDs to “TurnLightOn”, “TurnLightOff”, or “DimLight”. Thus, for radio A to turn off a light on radio B, radio A would send a transmission to the light switch endpoint on radio B, using cluster ID “TurnLightOff”. This is shown in the figure below.

© 2008 Digi International, Inc.

Figure 3-05. ZigBee Layer-Addressing Example



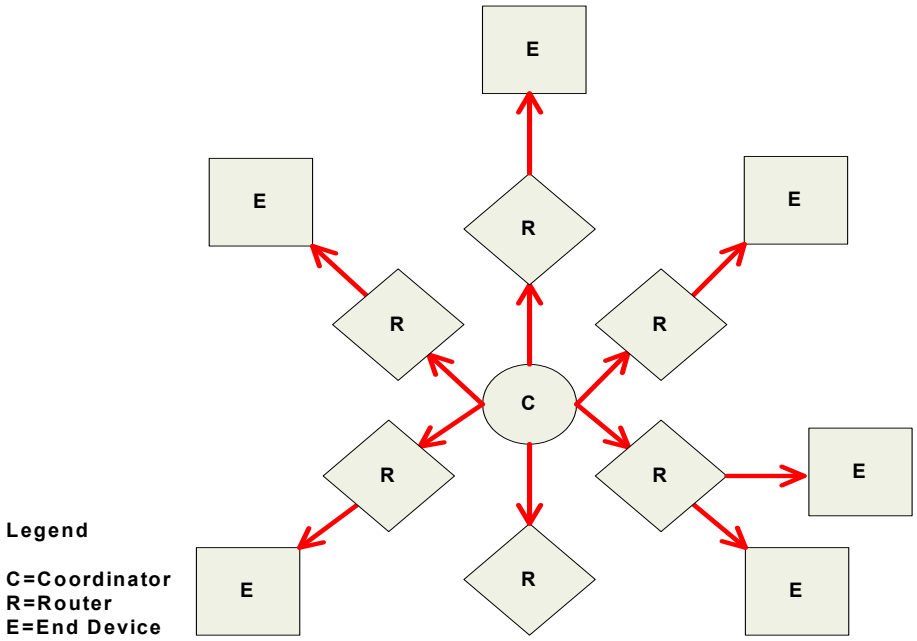
3.4.3. Data Transmission and Routing

All data packets are addressed using both device and application layer addressing fields. Data can be sent as a broadcast, or unicast transmission.

Broadcast Transmissions

Broadcast transmissions within the ZigBee protocol are intended to be propagated throughout the entire network such that all nodes receive the transmission. To accomplish this, all devices that receive a broadcast transmission will retransmit the packet 3 times.

Figure 3-06. Broadcast Data Transmission



Each node that transmits the broadcast will also create an entry in a local broadcast transmission table. This entry is used to keep track of each received broadcast packet to ensure the packets are

not endlessly transmitted. Each entry persists for 8 seconds. The broadcast transmission table holds 8 entries.

For each broadcast transmission, the ZigBee stack must reserve buffer space for a copy of the data packet. This copy is used to retransmit the packet as needed.Large broadcast packets will require more buffer space.

Since broadcast transmissions are retransmitted by each device in the network, broadcast messages should be used sparingly.

Unicast Transmissions

Unicast ZigBee transmissions are always addressed to the 16-bit address of the destination device. However, only the 64-bit address of a device is permanent; the 16-bit address can change. Therefore, ZigBee devices may employ network address discovery to identify the current 16-bit address that corresponds to a known 64-bit address, and route discovery to establish a route.

Network Address Discovery

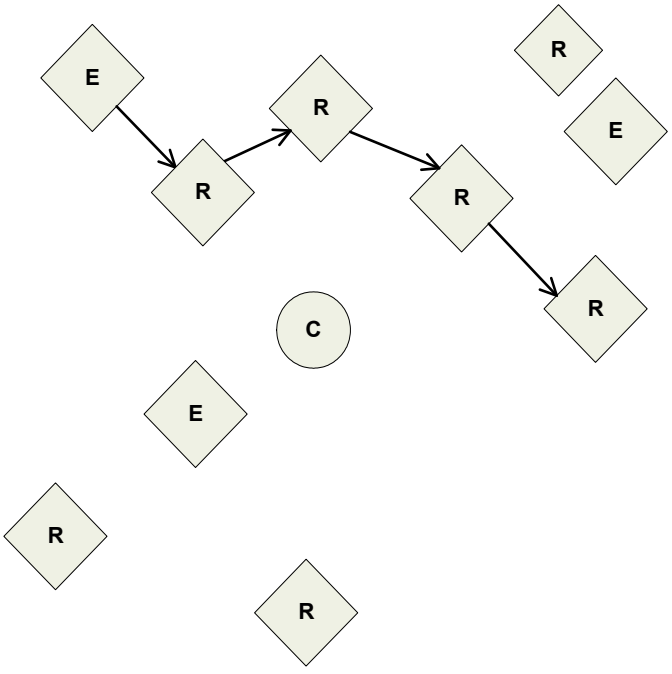
Data transmissions are always sent to the 16-bit network address of the destination device. However, since the 64-bit address is unique to each device and is generally known, ZigBee devices must discover the network address that was assigned to a particular device when it joined the PAN before they can transmit data.

To do this, the device initiating a transmission sends a broadcast network address discovery transmission throughout the network. This packet contains the 64-bit address of the device the initiator needs to send data to. Devices that receive this broadcast transmission check to see if their 64-bit address matches the 64-bit address contained in the broadcast transmission. If the addresses match, the device sends a response packet back to the initiator, providing the network address of the device with the matching 64-bit address. When this response is received, the initiator can then transmit data.

Route Discovery

ZigBee employs mesh routing to establish a route between the source device and the destination. Mesh routing allows data packets to traverse multiple nodes (hops) in a network to route data from a source to a destination. Routers and coordinators can participate in establishing routes between source and destination devices using a process called route discovery. The Route discovery process is based on the AODV (Ad-hoc On-demand Distance Vector routing) protocol.

Figure 3-07. Sample Transmission Through a Mesh Network



AODV (Ad-hoc On-demand Distance Vector) Routing Algorithm

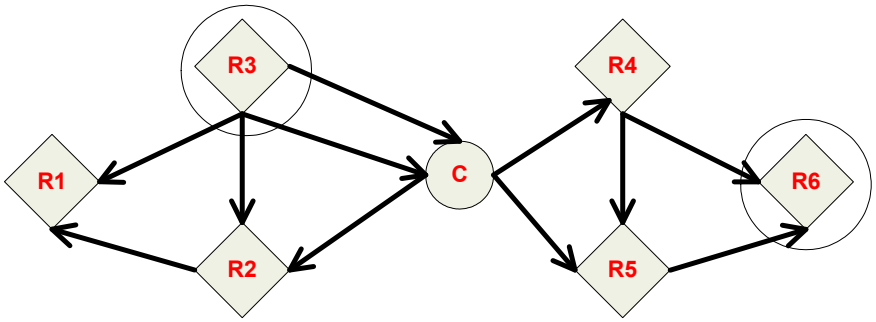
Routing under the AODV protocol is accomplished using tables in each node that store the next hop (intermediary node between source and destination nodes) for a destination node. If a next hop is not known, route discovery must take place in order to find a path. Since only a limited number of routes can be stored on a Router, route discovery will take place more often on a large network with communication between many different nodes.

Table 3-01.

Node	Destination Address	Next Hop Address
R3	Router 6	Coordinator
C	Router 6	Router 5
R5	Router 6	Router 6

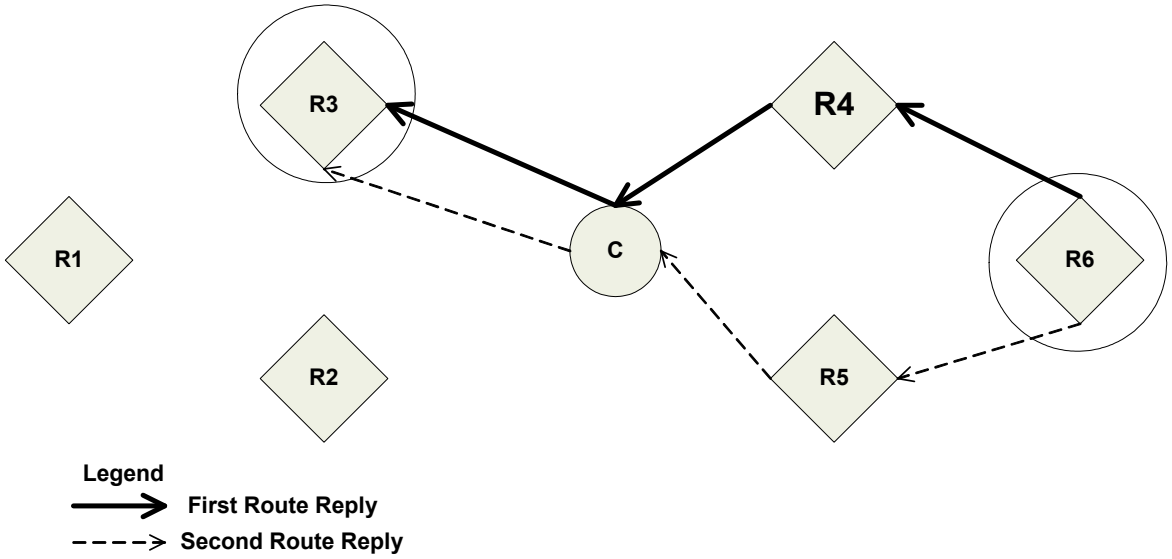
When a source node must discover a route to a destination node, it sends a broadcast route request command. The route request command contains the source network address, the destination network address and a path cost field (a metric for measuring route quality). As the route request command is propagated through the network (refer to the Broadcast Transmission), each node that re-broadcasts the message updates the path cost field and creates a temporary entry in its route discovery table.

Figure 3-08. Sample Route Request (Broadcast) Transmission Where R3 is Trying to Discover a Route to R6t



When the destination node receives a route request, it compares the 'path cost' field against previously received route request commands. If the path cost stored in the route request is better than any previously received, the destination node will transmit a route reply packet to the node that originated the route request. Intermediate nodes receive and forward the route reply packet to the source node (the node that originated route request).

Figure 3-09. Route Reply Sample Route Reply (Unicast) Where R6 Sends a Route Reply to R3.



Note: R6 could send multiple replies if it identifies a better route.

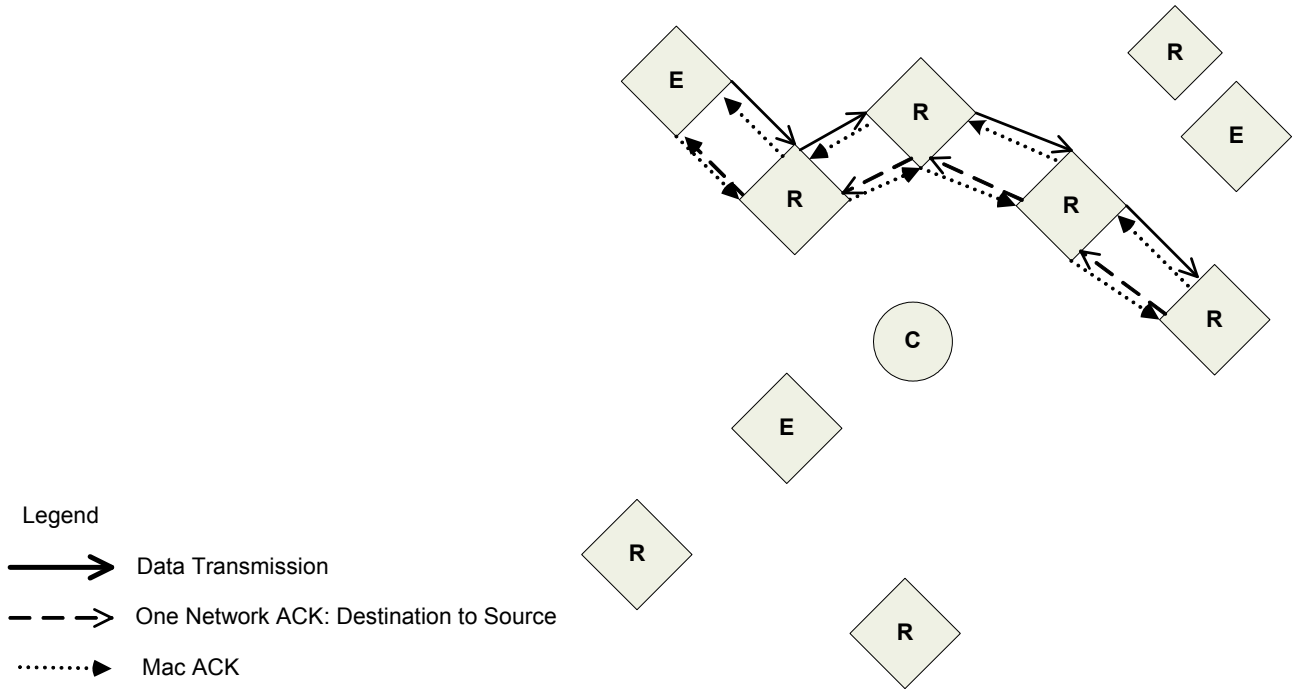
Retries and Acknowledgments

ZigBee includes acknowledgment packets at both the Mac and Application Support (APS) layers. When data is transmitted to remote device, it may traverse multiple hops to reach the destination. As data is transmitted from one node to its neighbor, an acknowledgment packet (Ack) is transmitted in the opposite direction to indicate that the transmission was successfully received. If the Ack is not received, the transmitting device will retransmit the data, up to 4 times. This Ack is called the Mac layer acknowledgment.

In addition, the device that originated the transmission expects to receive an acknowledgment packet (Ack) from the destination device. This Ack will traverse the same path that the data traversed, but in the opposite direction. If the originator fails to receive this Ack, it will retransmit the data, up to 2 times until an Ack is received. This Ack is called the ZigBee APS layer acknowledgment.

Refer to the ZigBee specification for more details.

Figure 3-010. Unicast Data Transmission



4. XBee ZNet 2.5 Networks

4.1. XBee ZNet 2.5 Network Formation

The XBee ZNet 2.5 modules are based off the EmberZNet 2.5.x software stack. This stack is similar in many instances to the ZigBee 2006 standard, but includes added features that are necessary to maintain a robust mesh network.

This chapter details how to configure the module networking, security, and addressing parameters to create operational mesh networks.

To create a ZigBee network, a coordinator must be started on a channel and PAN ID. Once the coordinator has started, routers and end device can join the network. Network formation is governed by the SC (Scan Channels), ID (PAN ID), SD (Scan Duration), and NJ (Node Join Time) commands. The SC and ID settings should be written using the WR command to preserve network formation or joining information.

4.1.1. Starting an XBee ZNet 2.5 Coordinator

In order to form a network, a coordinator must select an unused operating channel and PAN ID. The XBee ZNet 2.5 modules support user-settable commands to control the process of selecting a channel and PAN.

Table 4-02. Commands that affect coordinator channel and PAN ID selection. See the command table for details.

Command	Description
ID	Used to specify the PAN ID. Setting the ID to 0xFFFF allows any PAN ID to be selected.
SC	Selects a list of channels to scan for the energy and PAN scans.
SD	Determines the duration of time to scan on each of the SC channels for the energy and PAN scans.

If a coordinator has not already selected an operating PAN ID and channel, it uses the ID, SC, and SD parameter values to perform an energy scan and a PAN scan (see section 3.1.2). After completing these scans, the coordinator selects a valid channel and PAN ID and starts the PAN. If security is enabled (EE command), the coordinator will use the security key specified by KY.

If the coordinator is running API firmware, the API modem status "coordinator started" frame is sent out the Uart when the coordinator has started. It will then allow routers and end devices to join to it for a time based on the NJ parameter. The NJ parameter specifies the coordinator's permit-joining attribute value. (See section 3.1.4 and the command table for details.)

If the Associate LED function is enabled (D5 command), the Associate pin (pin 15) is configured in an output high state if the coordinator has not started. Once the coordinator selects a PAN ID and channel, the Associate pin toggles its output state 1x per second. (The toggle rate is settable using the LT command. The default behavior (LT=0) for a coordinator is one blink per second.)

Once the coordinator has selected a channel and PAN ID, it will retain the channel and PAN ID information through power cycle or reset events.

Example

1. Set SC and ID to the desired scan channels and PAN ID values. (The default values should suffice.)

2. If SC or ID is changed from the default, issue the WR command to write the new values.

3. The Associate LED, if enabled using the D5 (DIO5 Configuration) parameter, will start blinking 1x per second once the coordinator has started.

4. If API is enabled (AP parameter > 0): The API Modem Status "Coordinator Started" is sent out the UART.

5. The AI (Association Indication) parameter will be 0 signifying a successful startup.

6. The MY (16-bit Source address) attribute is 0 (the 16-bit network address of a ZigBee coordinator).

7. After startup, the coordinator will allow joining based on its NJ (Node Join Time) value.s

4.1.2. Joining an XBee ZNet 2.5 Router or End Device to a PAN

Before a router or an end device can participate in a ZigBee network, it must locate a nearby coordinator (or another router that has already joined a PAN) and attempt to join to it. The commands listed in table 4-02 also govern the joining behavior of the router / end device.

If a router or end device has not joined a ZigBee network, it performs a PAN scan on each of the SC channels, looking for a coordinator or router operating on a valid PAN ID that is allowing joins. The ID command on the router / end device defines a valid PAN ID. If ID=0xFFFF, any PAN ID is considered valid. For all other ID values, the router or end device will only join a network operating on the same PAN ID. The router / end device continues scanning the SC channels until it discovers a valid device to join.

Once the router / end device discovers a nearby device operating on a valid PAN ID, it sends an association request frame to that device, and receives an association response, indicating the router / end device was allowed onto the network. The joining router / end device then sends a broadcast discovery frame to discover the 64-bit address of the coordinator.

If the router or end device is running API firmware, the API modem status "joined" frame is sent out the Uart when it successfully joins a PAN. When a router joins, it will then allow routers and end devices to join to it for a time based on the NJ parameter. The NJ parameter specifies the router's permit-joining attribute value. (See section 3.1.4 and the command table for details.) End devices cannot allow other devices to join a PAN.

If the Associate LED function is enabled (D5 command), the Associate pin (pin 15) is configured in an output high state if the router / end device has not joined a PAN. Once the router / end device succeeds in joining a PAN, the Associate pin toggles its output state 2x per second. (The toggle rate is settable using the LT command. The default behavior (LT=0) for a router / end device is two blinks per second.)

Once the router / end device has joined on a channel and PAN ID, it will retain the channel and PAN ID information through power cycle or reset events. See section 4.1.6 for conditions that cause a router / end device to change its channel and/or PAN ID.

Each router and coordinator can allow up to 8 end device children to join. Once 8 end device children have joined a particular router or coordinator, that device cannot allow any more end devices to join to it until its network parameters are reset (see section 4.1.6). The number of end device children joined to a router or coordinator can be read with the NC command.

Example

1. Set SC and ID to the desired scan channels and PAN ID values. (The default values should suffice.)

2. If SC or ID is changed from the default, issue the WR command to write the new values.

3. The router, on power-up, will attempt to locate a parent to join according to its SC and ID parameters.

4. Once the router has joined a parent, the Associate LED, if enabled (D5), will start blinking 2x per second.

5. The OP and CH parameters will reflect the operating PAN ID and Channel.

6. The MY parameter will reflect the 16-bit network address of the router.

7. If API is enabled (AP parameter > 0): The API Modem Status "Joined" is sent out the UART.

8. If the router is not joining as expected, the AI (Association Indication) parameter can be read to determine the cause of failure.

9. Once the router has joined a PAN, the router will allow joining based on the NJ parameter.

© 2008 Digi International, Inc.

4.1.3. Verifying Network Selection

The AI command can be used to determine the status of the last join attempt. For example, if AI=0x22, it means the router or end device found PANs, but none were operating on the right PAN ID. See the AI command entry in the AT Command table for details.

The actual operating channel and PAN ID that the device operates on can be read with the CH and OP commands, respectively. If ID is not equal to 0xFFFF (join any PAN ID), the OP and ID commands will return the same value. The MY command indicates the device's 16-bit address.

4.1.4. Secure Networks

If security is enabled (EE=1), the coordinator will startup using the 128-bit AES encryption key specified by the KY command. Routers and end devices that join the PAN must either be configured with the same security key, or they must obtain the key over the air when they join. The security key can be configured on routers and end devices using serial commands or by sending remote API commands when commissioning devices. As an alternate, but less secure approach, the security key can be transmitted over-the-air during joining if the EO (encryption options) command is set to allow sending the key over-the-air when joining.

Security Policies

The EO command is used to define the security policy. In XBee ZNet 2.5, the security policy defines what types of devices can join the network, and who makes the decision regarding which devices can join.

If EO is set to send the security over the air during joining, the network will allow devices to join that do or do not already have the security key. When a device joins this type of network, if it does not have the security key, it will be transmitted to that device over the air unencrypted (in the clear) when it joins. When the device receives the security key, it will use the key to encrypt and decrypt all future data transmissions.

If EO is set to enable a trust center, all join requests are sent to the trust center for verification. In XBee ZNet 2.5, the trust center is always the coordinator. The coordinator will determine whether or not to allow joins based on its EO setting, and whether or not the joining device has the security key. If EO on the coordinator is set to send the security key over the air during joining, the coordinator will allow devices that do or do not already have the security key to join. Otherwise, the coordinator will only allow devices to join the network that already have the encryption key.

Data Encryption

When encryption is enabled in the network, RF data packets are encrypted and decrypted at the network layer according to AES-128 (FIPS 197) algorithm. The following packet components are encrypted:

- Network payload
- APS headers
- Data payload.

For multi-hop transmissions, each router along the route must decrypt and re-encrypt the data. Enabling security in a network increases the latency of data transmissions. In addition, enabling security adds several bytes of overhead into the RF packet. With security enabled, up to 65 bytes of data (payload) can be sent in a single packet.

4.1.5. Open and Closed Networks

In addition to setting the permit-joining attribute on the coordinator and routers, the NJ command also defines an open or closed network. An open network is a network where joining is always enabled (NJ=0xFF on all devices). This type of network should be used to support:

- end devices that may change parents (i.e. – end devices that move from one location to another).
- changing the operating channel of an entire PAN

A network is considered closed if NJ < 0xFF (permit-joining attribute enables joining for a finite time). If NJ < 0xFF, joining may or may not be permitted. Closed networks should only be used if

a network is static – end devices are static and should not change their parent, and the operating channel should not be changed.

In most cases, NJ should be set the same on all devices (coordinator, routers, and end devices) in a PAN.

Open Network End Device Parent Link Quality

In an open network, end devices will leave their existing parent and attempt to rejoin the PAN if the end device's parent fails to send an acknowledgment after three consecutive transmissions or poll requests.

Open Network Channel Verification

In an open network, routers validate the channel (if JV=1) when joining or after a power cycle or reset event. To do this, they send a unicast transmission to the coordinator, to discover the coordinator's 64-bit address. If the coordinator doesn't respond to multiple discovery attempts, the router leaves the PAN and attempts to join a new PAN, based on its ID and SC parameters. If JV=0 or if the network is a closed network (NJ < 0xFF), the router will not leave its current channel if a coordinator is not found.

If channel verification is enabled, the Associate LED will not blink on the router until the coordinator has responded.

4.1.6. Resetting Network Parameters

Once a coordinator has started, or a router or end device has joined, the device will continue operating on the same channel and PAN ID (even through power cycle or reset) until one of the following occurs:

- The PAN ID changes (ID command) such that the current operating PAN ID is invalid
- The scan channels mask changes (SC command) such that the current operating channel is not valid
- One of the security parameters is changed (EE, EO, KY)
- After a reset or power cycle, in an open network, a router does not receive an address discovery response from the coordinator
- In an open network, an end device does not receive an acknowledgment from its parent for three consecutive transmissions
- The NR0 command is issued to force a device to leave.

If any of the above conditions occur on a coordinator, the coordinator will attempt to start a new PAN based on the SC and ID parameters as described in section 4.1.1. If any of the above conditions occur on a router or end device, it will leave its current PAN and attempt to join a new PAN based on its ID and SC parameters.

Note that with the exception of NR0, any command parameter changes do not take effect until the changes are applied.

4.2. XBee ZNet 2.5 Addressing

XBee modules support both ZigBee device addressing and application-layer addressing. Device addressing provides a simple means of sending data from one device to another by hiding the application layer addressing information (ZigBee endpoints and cluster IDs). If a device will support multiple endpoints or cluster IDs, application addressing can be used to include endpoint and cluster ID information in the transmission.

Device addressing supports transmissions to:

- a destination 64-bit address
- a destination NI-string
- the ZigBee PAN coordinator
- all devices on the PAN (broadcast).

Application-layer addressing supports transmissions to:

- all of the above device addressing destinations
- specific endpoints on a destination device
- specific cluster IDs on a destination device.

4.2.1. Device Addressing

All XBee ZNet 2.5 modules can be identified by their unique 64-bit addresses or a user-configurable ASCII string identifier The 64-bit address of a module can be read using the SH and SL commands. The ASCII string identifier is configured using the NI command. To transmit using device addressing, only the destination address must be configured. The destination address can be specified using either the destination device's 64-bit address or its NI-string. The XBee modules also support coordinator and broadcast addressing modes. Device addressing in the AT firmware is configured using the DL, DH, or DN commands. In the API firmware, the ZigBee Transmit Request API frame (0x10) can be used to specify destination addresses.

64-Bit Addressing (Transparent)

To address a node by its 64-bit address, the destination address must be set to match the 64-bit address of the remote. In the AT firmware, the DH and DL commands set the destination 64-bit address. In the API firmware, the destination 64-bit address is set in the ZigBee Transmit Request frame.

To send a packet to an RF module using its 64-bit Address (Transparent Mode)

Set the DH (Destination Address High) and DL (Destination Address Low) parameters of the source node to match the 64-bit Address (SH (Serial Number High) and SL (Serial Number Low) parameters) of the destination node

Since the ZigBee protocol relies on the 16-bit network address for routing, the 64-bit address must be converted into a 16-bit network address prior to transmitting data. If a module does not know the 16-bit network address for a given 64-bit address, it will transmit a broadcast network address Discovery command. The module with a matching 64-bit address will transmit its 16-bit network address back. Once the network address is discovered, the data will be transmitted.

The modules maintain a table that can store up to seven 64-bit addresses and their corresponding 16-bit network addresses.

64-bit Addressing (API)

To send a packet to an RF module using its 64-bit Address (API Mode)

- Use the ZigBee Transmit Request API frame to set 64-bit destination address of the source node to match the 64-bit Address (SH (Serial Number High) and SL (Serial Number Low) parameters) of the destination node.
- If the 16-bit address of the destination node is not known, set 16-bit destination network address to 0xFFFE.

To send an API transmission to a Coordinator using its 16-bit network address:

- Set the 64-bit Destination Address field to all 0's.

API Mode provides the ability to store and maintain 16-bit network address tables on an external processor. The 16-bit network address information is provided to the application through the following:

- The ZigBee Transmit Status Frame
  (contains the current 16-bit network address of the remote)
- The ND and DN commands
  (return 64-bit and 16-bit network addresses of remote nodes)

With this information, a table can be built in an application that maps a 64-bit Address to the corresponding 16-bit network address.

The ZigBee Transmit Request API frame specifies the 64-bit Address and the network address (if known) that the packet should be sent to. By supplying both addresses, the module will forego network address Discovery and immediately attempt to route the data packet to the remote. If the network address of a particular remote changes, network address and route discovery will take place to establish a new route to the correct node. Upon successful packet delivery, the TX Status Frame will indicate the correct network address of the remote.

Table 4-03. Sample table mapping 64-bit Addresses to 16-bit Network Addresses

Index	64-bit Address	16-bit Network Address
0	0013 A200 4000 0001	1234
1	0013 A200 4000 0002	5678
2	0013 A200 4000 01A0	A479
3	0013 A200 4000 0220	FFFE (unknown)

NI-String Addressing

The NI string can alternatively be used to address a remote module.

To send a packet to an RF module using its NI-string (Transparent Mode)

Issue the DN (Destination Node) command using the NI (Node Identifier)-string of the destination node as the parameter.

To send a packet to an RF module using its NI-string (API Mode)

Issue the DN command as stated above using the AT Command API frame.

When the DN command is issued, a broadcast transmission is sent across the network to discover the module that has a matching NI (Node Identifier) parameter. If a module is discovered with a matching NI-string, the DH and DL parameters will be configured to address the destination node and the command will return both the 64-bit Address and the 16-bit network address of the discovered node. Data can be transmitted after the DN (Destination Node) command finishes.

the AO command. See “API Frames” section for details.

Coordinator Addressing

A Coordinator can be addressed using its 64-bit address or NI string as described in the “NI-String Addressing” section. Alternatively, since the ZigBee Coordinator has a network address of “0”, it can be addressed by its 16-bit network address.

To send a transmission to a Coordinator using its 16-bit network address:

Set the Destination Address of the transmitting module as shown below:

AT Firmware

DL (Destination Low Address) = 0

DH (Destination High Address) = 0

API Firmware

Set the 64-bit destination address field in the API transmit frame to

0x0000000000000000.

Broadcast Addressing

Broadcast transmissions are sent using a 64-bit address of 0x0000FFFF. Any RF module in the PAN will accept a packet that contains a broadcast address. When configured to operate in Broadcast Mode, receiving modules do not send ACKs (Acknowledgements).

To send a broadcast packet to all modules

Set the Destination Addresses of the transmitting module as shown below:

AT Firmware

DL (Destination Low Address) = 0x0000FFFF
DH (Destination High Address) = 0x00000000

API Firmware

Set the 64-bit destination address field in the API transmit frame to 0x000000000000FFFF.

NOTE: When programming the module, parameters are entered in hexadecimal notation (without the "0x" prefix). Leading zeros may be omitted.

Refer to the "Broadcast Transmissions" for more information.

4.2.2. Application-layer Addressing

Application-layer addressing allows the application to specify endpoint and cluster ID values for each transmission. Addressing multiple endpoints and cluster IDs can be accomplished by explicitly setting these values as needed.

In AT firmware, application-layer addressing must be enabled using the ZA command. When application-layer addressing is enabled, the DE and SE commands specify the source and destination endpoints, and the CI command sets the cluster ID that will be used in the transmission.

In API firmware, the Explicit Addressing ZigBee Command frame (0x11) can be used to configure the endpoint and cluster ID addressing parameters as needed. The destination device can indicate application-layer addressing information depending on the AO parameter. Some endpoint values are reserved for use by the XBee ZNet 2.5 modules and should not be used. Please refer to the following tables for these values:

Table 4-04. XBee ZNet 2.5 Endpoint allocation

Endpoint	Description
0	ZigBee Device Objects endpoint. Reserved for ZigBee stack.
0x01 - 0xDB (219)	Available endpoints
0xDC (220) - 0xEE (238)	Reserved for Digi Use
0xE6 (230)	Command Endpoint
0xE8 (232)	Data Endpoint
0xEF (239) - 0xF0 (240)	Reserved for Ember Use

© 2008 Digi International, Inc.

The following cluster IDs are supported on the data endpoint (0xE8):

Table 4-05. Supported Cluster IDs on the Data Endpoint

Cluster ID	Name	Description
0x11	Transparent serial data	This is the default cluster ID used to transmit serial data.
0x12	Serial loopback data	Data received on this cluster ID is transmitted back to the sender.
0x92	IO sample data	IO samples are transmitted to a remote on this cluster ID.
0x94	XBee sensor sample data	An XBee sensor device transmits sensor readings on this cluster ID.
0x95	Node identification	A single press on the commissioning button sends a broadcast transmission to this cluster ID.

4.3. Sleeping End Devices

XBee ZNet 2.5 modules support sleep mode operation in the Router / End Device firmware. Sleep modes allow a ZigBee end device to enter a low power mode when idle and wake as needed to transmit or receive data.

End devices must join to a router or coordinator to become part of a network. When the join occurs, the end device becomes the child of the router or coordinator that allowed the join, and the device that allowed the join becomes the end device's parent.

A router or coordinator can only allow up to 8 end devices to join to it. Once 8 end devices have joined to a parent, no additional end devices can join until a network reset condition occurs on the parent.

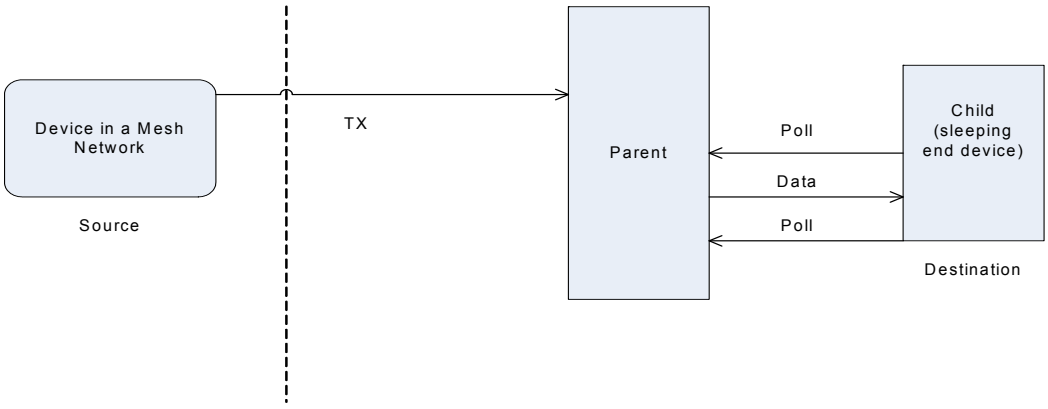
4.3.1. End Device Operation

ZigBee end devices rely on a parent (router or coordinator) to remain awake and receive any data packets destined for the end device. When the end device wakes from sleep, it sends a transmission (poll request) to its parent asking if the parent has received any RF data destined for the end device. The parent, upon receipt of the poll request, will send an RF response and the buffered data (if present).

If the parent has no data for the end device, the end device may return to sleep, depending on its sleep mode configuration settings. The following figure demonstrates how the end device uses polling to receive RF data through its parent.

If the end device is awake with the ST timer running (SM=4), or if Sleep_RQ is de-asserted (SM=1), the end device will send poll requests every 100ms to ensure it receives any new RF data from its parent.

Figure 4-011.RF Data Sent to Sleeping End Device



When RF data is sent to a sleeping end device, the end device's parents buffers the data until the end device polls for the data, or a timeout occurs

4.3.2. Parent Operation

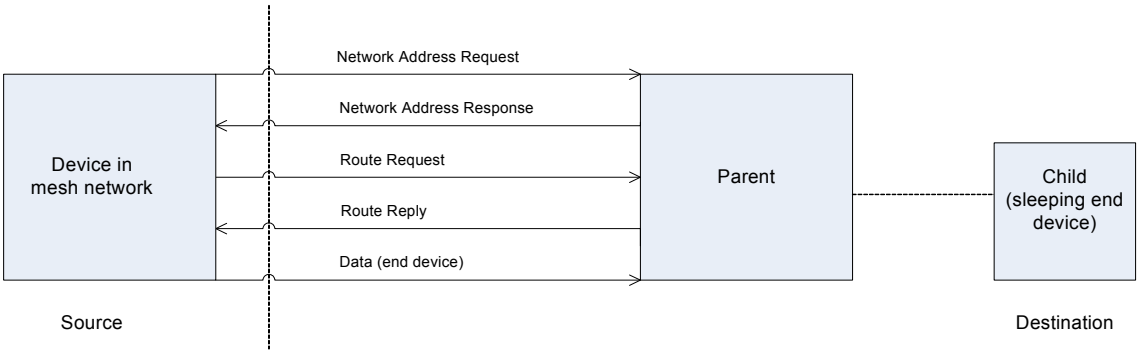
When an end device joins a ZigBee PAN, it becomes a child of the (coordinator or router) device it joined to, and the device that allowed the join becomes the end device's parent. Thereafter, the parent will manage RF data packets for the end device. If the parent receives an RF packet destined for the end device, it will store the data packet until one of the following occurs:

- The parent runs out of storage space and cannot store a new packet.
- A packet has been stored for a period of time (timeout).
- The destination end device child sends a poll request transmission to request the data packet.

When the parent stores a packet destined for an end device child, it stores the packet for a maximum time set by SP. The actual storage time is computed as (SP * 2.5), not exceeding 30 seconds. If end devices implement cyclic sleep, SP should be set the same on a parent as it is on their sleeping end device children. In the case of pin sleep, where RF data could be received, the end device should wake within SP time to ensure incoming RF data is not lost. The parent can only store one broadcast packet (the most recently received) for its end device children.

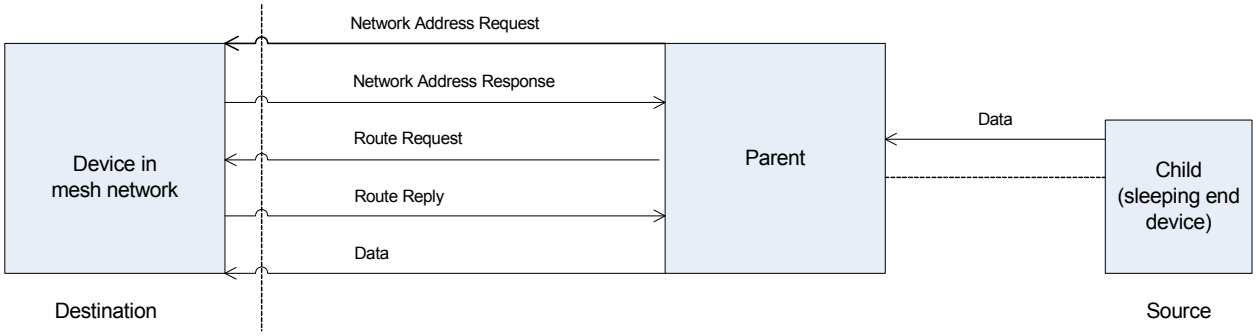
The parent is also responsible for performing any route or address discoveries to forward data sent by its end device child(ren) into the mesh network. The parent's interactions with the mesh network in behalf of its end device child(ren) are shown in the figure below. (Note address and route discoveries occur only as needed.)

Figure 4-012. Determining the End Device Route



To talk to an end device, a source device in a mesh network must talk to the parent to determine the address and the route of the end device. The source device then sends data to the parent.

Figure 4-013.End Device: Sending Data to Parente



An end device sends data to its parent to route to the correct destination node. The parent performs all of the necessary addressing and route discoveries before forwarding the data.

4.3.3. End Device Behavior

An end device child retrieves RF data from its parent through polling. When the end device wakes from sleep, it sends a poll request to its parent, notifying the parent that it is awake, and requesting any data the parent has received that was addressed to the end device child. If the parent has data for the end device, the end device will continue polling.

4.3.4. Parent Behavior

The parent of an end device remains awake and can receive data packets intended for any of its end device children. Since the end device child may be sleeping, the parent buffers any received RF data that is destined for an end device child until the end device asks for it, or until a timeout occurs. This timeout is settable using the SP command. The actual timeout is calculated as (2.5 * SP), not exceeding 30 seconds.

4.3.5. End Device Sleep Configuration

Configuration parameters exist to customize the mechanisms for entering sleep and defining sleep and wake times. The XBee ZNet 2.5 modules support both pin sleep and cyclic sleep modes. The sleep mode is settable using the SM command. If SM=0, sleep mode is disabled and the device operates as a router. If SM changes from 0 to a non-zero value, the router leaves the ZigBee network and attempts to rejoin as an end device. For this change to be successful, the end device must be able to join a nearby router or coordinator that is allowing end device joins. (See [Add Reference Here] for details.)

The On/Sleep pin (pin 13) provides a hardware indication of whether the module is asleep or not. On/Sleep is de-asserted (low) when the module enters sleep and asserted (high) when the module wakes.

If $\overline{\text{CTS}}$ flow control is disabled (D7 command), the $\overline{\text{CTS}}$ pin is also de-asserted (high) when entering sleep, and asserted (low) upon waking.

Pin Sleep

Pin sleep puts the module to sleep and wakes it from sleep according to the state of Sleep_RQ (pin 9). Pin sleep is enabled by setting SM to 1.

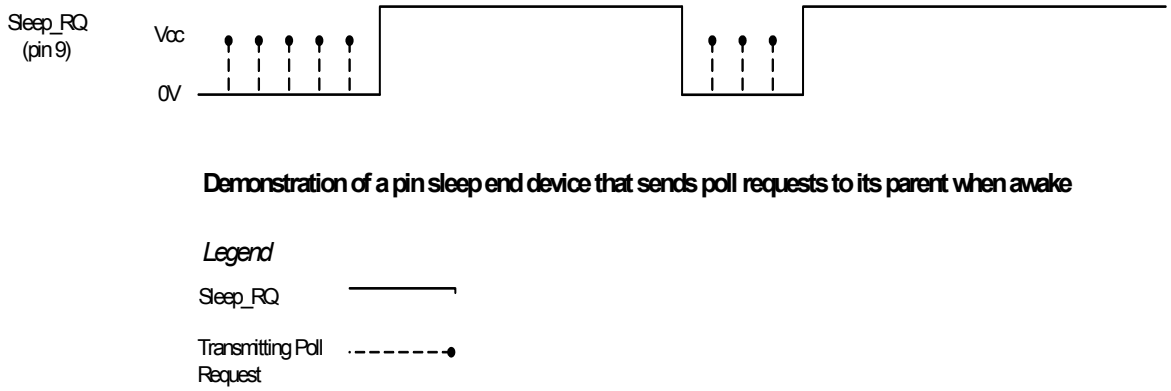
When Sleep_RQ is asserted (high), the module will finish any transmit or receive operations, and then enter a low power state. If the module has not joined a network and Sleep_RQ is asserted,

the module will sleep once the current join attempt completes (ie scanning for a valid network to join). While asleep, the module will not respond to serial or RF activity.

To wake a module operating in pin sleep, de-assert Sleep_RQ (pin 9). The module will wake when Sleep_RQ is de-asserted and is ready to transmit or receive when the CTS line is low. If the module has not joined a network, it will scan for a network to join when it wakes.

When a joined end device wakes from pin sleep, it sends a poll request to its parent to see if the parent has buffered data for the end device. The end device will continue to send poll requests every 100ms while it remains awake.

Figure 4-014. Demonstration of Pin Sleep

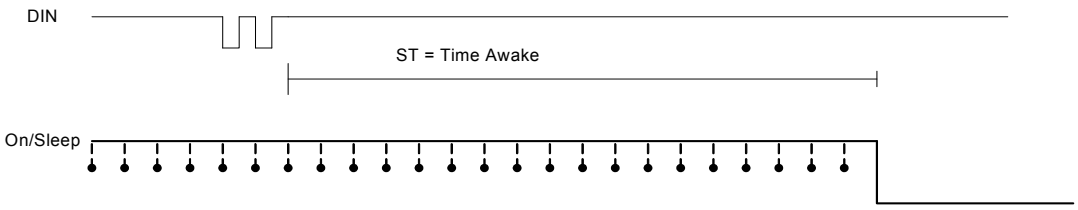


Cyclic Sleep

Cyclic sleep allows modules to wake periodically to check for RF data and sleep when idle. When the SM parameter is set to 4 or 5, the module operates in cyclic sleep mode. Setting SM to 5 allows the module to be awakened from sleep on a high-to-low transition on Sleep_RQ (pin 9). Setting SM to 4 disables the pin wake option.

In cyclic sleep mode, if serial or RF data is received, the module will start an inactivity timer and remain awake until this timer expires. The inactivity time is settable with the ST command. While the module is awake, it will continue to send poll request transmissions to its parent to check for buffered data every 100ms. The timer will be restarted anytime serial or RF data is received. The module will resume sleep when the timer expires. This behavior is shown in the following figure.

Figure 4-015.Cylic Sleep



A cyclic sleep end device enters sleep mode when no serial or RF data is received for ST time.

Legend

On/Sleep

Transmitting Poll Request

In cyclic sleep, the time the module sleeps for is dependent on several configuration commands – SP, SN, and SO.

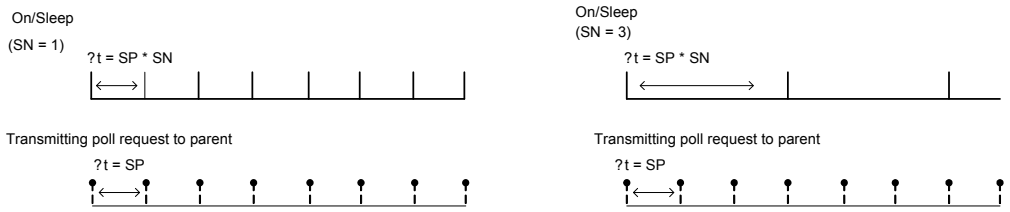
With SO set to 0 (no sleep options - default), the sleep period is defined as (SP * SN). Since the parent can only buffer data up to 30 seconds, SP is settable up to 28 seconds. After SP expires, the module wakes to send a poll request transmission to the parent to check for data. This helps ensure the end device can receive RF data transmissions that were sent to it.

In many cases, the On/Sleep pin can be used to wake an external microprocessor or peripheral device when the module wakes from sleep. If the end device wakes and finds that its parent had no data, there may be no need to wake the external device. The SN command is a multiplier of the SP time that determines how often to set the On/Sleep pin when waking. For example, if the end device sleeps for 20 seconds, but the On/Sleep pin should only be set high on every 3rd poll (once per minute), SN can be set to 3. The On/Sleep pin will be set high anytime RF data is received, regardless of SN. This is shown in the figure below.

In some applications, the end device may transmit data at a very slow rate (once an hour, once a day, etc) and will only receive data in response to its transmission. In such cases, the SO command can be used to cause an end device to sleep for the entire SP * SN duration. This is shown below.

© 2008 Digi International, Inc.

Figure 4-016. Polling for Data without Asserting On/Sleep



Setting SN > 1 allows the XBee to silently poll for data without asserting On/Sleep. If RF data is received when polling, On/Sleep will immediately assert.

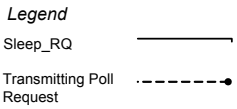
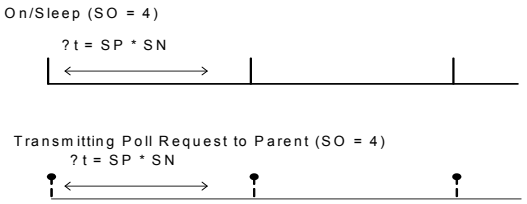


Figure 4-017. Transmitting Poll Request to Parent



Setting SO = 4 will cause the end device to sleep for the entire SP * SN time. This should only be used if the end device will not receive RF data while sleeping.

Since a parent can only buffer data up to 30 seconds, an end device should only sleep for more than 30 seconds if it will not receive RF data when sleeping. The ST parameter can be set to keep the end device awake after transmitting data to receive RF data. The commissioning push button can be used to force the end device to wake for 30 seconds. See section 6.4 for details.

Transmitting Data to End Devices

To reliably transmit data to an end device, SP should be set the same on the end device, the end device's parent, and on the device that is initiating a transmission to the end device. SP determines the transmission timeout on the sender, the time to buffer the received packet on the parent, and the time to sleep on the end device.

4.4. Remote Configuration Commands

The API firmware has provisions to send configuration commands to remote devices using the Remote Command Request API frame (see Chapter 7 – API Operation). This API frame can be used to send commands to a remote module to read or set command parameters.

The API firmware has provisions to send configuration commands (set or read) to a remote module using the Remote Command Request API frame (see chapter 8). Remote commands can be issued to read or set command parameters on a remote device.

4.4.1. Sending a Remote Command

To send a remote command, the Remote Command Request frame should be populated with the 64-bit address and the 16-bit address (if known) of the remote device, the correct command options value, and the command and parameter data (optional). If a command response is desired, the Frame ID should be set to a non-zero value.

4.4.2. Applying Changes on Remote

When remote commands are used to change command parameter settings on a remote device, parameter changes do not take effect until the changes are applied. For example, changing the BD parameter will not change the actual serial interface rate on the remote until the changes are applied. Changes can be applied using remote commands in one of three ways:

- Set the apply changes option bit in the API frame
- Issue an AC command to the remote device
- Issue a WR + FR command to the remote device to save changes and reset the device.

4.4.3. Remote Command Responses

If the remote device receives a remote command request transmission, and the API frame ID is non-zero, the remote will send a remote command response transmission back to the device that sent the remote command. When a remote command response transmission is received, a device sends a remote command response API frame out its UART. The remote command response indicates the status of the command (success, or reason for failure), and in the case of a command query, it will include the register value.

The device that sends a remote command will not receive a remote command response frame if:

- The destination device could not be reached
- The frame ID in the remote command request is set to 0.

4.5. IO Line Monitoring

XBee ZNet 2.5 modules support analog inputs and digital IO. Analog and digital IO can be set or read. The XBee ZNet 2.5 supports the following IO functions:

Table 4-06.

Module Pin Names	Module Pin Numbers	Configuration Command
CD/DIO12	4	P2
PWM0/RSSIM/DIO10	6	P0
PWM/DIO11	7	P1
SLEEP_RQ/DIO8	9	IO Configuration not supported
DIO4	11	D4
CTS/DIO7	12	D7
ON_SLEEP/DIO9	13	IO Configuration not supported

© 2008 Digi International, Inc.

Table 4-06.

Module Pin Names	Module Pin Numbers	Configuration Command
ASSOC/DIO5	15	D5
RTS/DIO6	16	D6
AD3/DIO3	17	D3
AD2/DIO2	18	D2
AD1/DIO1	19	DI
AD0/DIO0	20	D0

Setting the configuration command that corresponds to a particular pin will configure the pin. IO line command settings include the following:

Table 4-07.

Pin Command Parameter	Description
0	Unmonitored digital input
1	Reserved for pin-specific alternate functionalities
2	Analog input, single ended (A/D pins only)
3	Digital input, monitored
4	Digital output, default low
5	Digital output, default high
6-9	Alternate functionalities, where applicable

For example, sending the command "ATD23" will configure AD2/DIO2 (pin 18) as a digital input.

Pullup resistors can be set for each digital input using the PR command.

4.5.1. IO Samples

When an IO sample is taken, the collected data is assembled into a packet and either sent out the uart or transmitted to a remote device. The IO sample is formatted in the following manner:

Table 4-08.

Bytes	Name	Description
1	Sample Sets	Number of sample sets in the packet. (Always set to 1.)
2	Digital Channel Mask	Indicates which digital IO lines have sampling enabled. Each bit corresponds to one digital IO line on the module. • bit 0 = AD0/DIO0 • bit 1 = AD1/DIO1 • bit 2 = AD2/DIO2 • bit 3 = AD3/DIO3 • bit 4 = DIO4 • bit 5 = ASSOC/DIO5 • bit 6 = RTS/DIO6 • bit 7 = CTS/GPIO7 • bit 8 = N/A • bit 9 = N/A • bit 10 = RSSI/DIO10 • bit 11 = PWM/DIO11 • bit 12 = CD/DIO12 For example, a digital channel mask of 0x002F means DIO0,1,2,3, and 5 are enabled as digital IO.
1	Analog Channel Mask	Indicates which lines have analog inputs enabled for sampling. Each bit in the analog channel mask corresponds to one analog input channel. • bit 0 = AD0/DIO0 • bit 1 = AD1/DIO1 • bit 2 = AD2/DIO2 • bit 3 = AD3/DIO3 • bit 7 = Supply Voltage
Variable	Sampled Data Set	A sample set consisting of 1 sample for each enabled ADC and/or DIO channel. If any digital IO lines are enabled, the first two bytes of the data set indicate the state of all enabled digital IO. Only digital channels that are enabled in the Digital Channel Mask bytes have any meaning in the sample set. If no digital IO are enabled on the device, these 2 bytes will be omitted. Following the digital IO data (if any), each enabled analog channel will return 2 bytes. The data starts with AIN0 and continues sequentially for each enabled analog input channel up to AIN3, and the supply voltage (if enabled) at the end.

The sampled data set will include 2 bytes of digital IO data only if one or more IO lines on the device are configured as digital IO. If no pins are configured as digital IO, these 2 bytes will be omitted.

The digital IO data is only relevant if the same bit is enabled in the digital IO mask as shown in the following figure:

Analog samples are returned as 10-bit values. The analog reading is scaled such that 0x0000 represents 0V, and 0x3FF = 1.2V. (The analog inputs on the module cannot read more than 1.2V.) Analog samples are returned in order starting with AIN0 and finishing with AIN3, and the supply voltage. Only enabled analog input channels return data as shown in the figure below.

To convert the A/D reading to mV, do the following:

AD(mV)= (ADIO reading/0x3FF)*1200mV

The reading in the sample frame represent voltage inputs of 1144.9 and 342.5mV for ADIO0 and ADIO1 respectively.

© 2008 Digi International, Inc.

4.5.2. Queried Sampling

The IS command can be sent to a device locally, or to a remote device using the API remote command frame (see Chapter 8 for details). When the IS command is sent, the receiving device samples all enabled digital IO and analog input channels and returns an IO sample. If IS is sent locally, the IO sample is sent out the uart. If the IS command was received as a remote command, the IO sample is sent over-the-air to the device that sent the IS command.

If the IS command is issued in AT firmware, the module returns a carriage return-delimited list containing the above-listed fields. The API firmware returns an AT command response packet with the IO data included in the command data portion of the response frame.

The following table shows an example of the fields in an IS reponse.

.

Table 4-09.

Example	Sample AT Response
0x01	[1 sample set]
0x0C0C	[Digital Inputs: DIO 2, 3, 10, 11 low]
0x03	[Analog Inputs: A/D 0, 1]
0x0408	[Digital input states: DIO 3, 10 high, DIO 2, 11 low]
0x03D0	[Analog input ADIO 0= 0x3D0]
0x0124	[Analog input ADIO 1=0x120]

4.5.3. Periodic IO Sampling

Periodic sampling allows an XBee / XBee-PRO module to take an IO sample and transmit it to a remote device at a periodic rate. The periodic sample rate is set by the IR command. If IR is set to 0, periodic sampling is disabled. For all other values of IR, data will be sampled after IR milliseconds have elapsed and transmitted to a remote device. The DH and DL commands determine the destination address of the IO samples. DH and DL can be set to 0 to transmit to the coordinator, or to the 64-bit address of the remote device (SH and SL). Only devices running API firmware can send IO data samples out their Uart. Devices running AT firmware will discard received IO data samples.

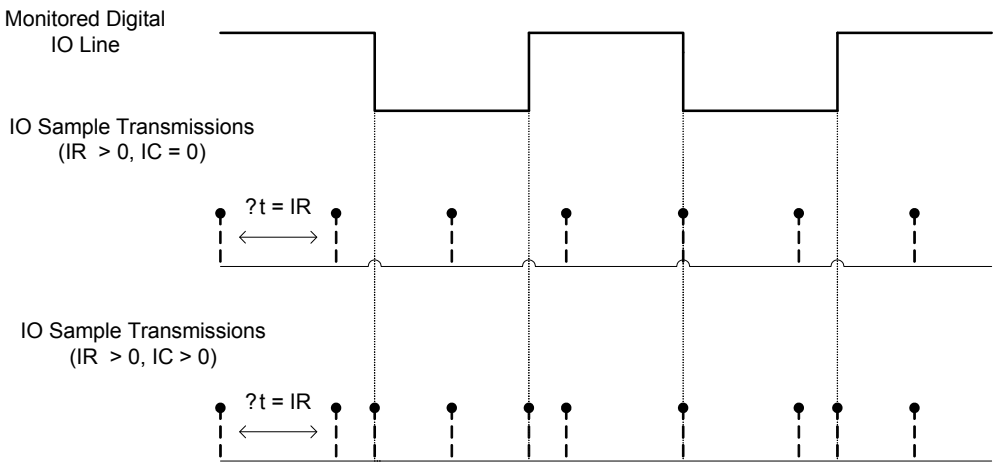
A sleepy end device will transmit periodic IO samples at the IR rate until the ST timer expires and the device can resume sleeping. See section 5.3 for more information on sleep.

4.5.4. Digital IO Change Detection

Modules can be configured to transmit a data sample immediately whenever a monitored digital IO pin changes state. The IC command is a bitmask that can be used to set which digital IO lines should be monitored for a state change. If one or more bits in IC is set, an IO sample will be transmitted as soon as a state change is observed in one of the monitored digital IO lines. Figure xx below shows how edge detection can work with periodic sampling.

© 2008 Digi International, Inc.

Figure 4-018. Enabling Edge Detection



Enabling Edge Detection will force an immediate sample of all monitored digital IO lines if any digital IO lines change state.

4.5.5. Voltage Supply Monitoring

The voltage supply threshold is set with the V+ command. If the measured supply voltage falls below or equal to this threshold, the supply voltage will be included in the IO sample set. V+ is set to 0 by default (do not include the supply voltage).

4.6. I/O Line Configuration

The XBee ZNet 2.5 modules support both analog input and digital IO line modes on several configurable pins.

Configuring A/D and Digital Lines

The following table lists the pin functions supported on the modules

.

Table 4-010.

Module Pin Names	Module Pin Numbers	Configuration Command
CD/DIO12	4	P2
PWM0/RSSI/DIO10	6	P0
PWM/DIO11	7	P1
SLEEP_RQ/DIO8	9	IO Configuration not supported
DIO4	11	D4
CTS/DIO7	12	D7
ON_SLEEP/DIO9	13	IO Configuration not supported
ASSOC/DIO5	15	D5
RTS/DIO6	16	D6
AD3/DIO3	17	D3
AD2/DIO2	18	D2

© 2008 Digi International, Inc.

Table 4-010.

Module Pin Names	Module Pin Numbers	Configuration Command
AD1/DIO1	19	DI
AD0/DIO0	20	D0

Setting the configuration command that corresponds to a particular pin will configure the pin. Parameters for the pin configuration commands typically include the following:

.

Table 4-011.

Pin Command Parameter	Description
0	Unmonitored digital input
1	Reserved for pin-specific alternate functionalities
2	Analog input, single ended (A/D pins only)
3	Digital input, monitored
4	Digital output, default low
5	Digital output, default high
6-9	Alternate functionalities, where applicable

See the command table for more information. Pullup resistors for each digital input can be enabled using the PR command.

Sampling A/D and Digital Input Lines

The IS command can be used to sample the current value of all enabled A/D and digital input lines. .

Table 4-012.

Bytes	Name	Description
1	Sample sets in packet	Number of sample sets in the packet
2	Digital Channel Mask	Each bit in the digital channel mask corresponds to one digital IO line. The bits, from LSB to MSB, correspond to DOI0-DOI5 on the module. For example a digital channel mask of 0x002F means DIO0,1,2,3, and 5 are enabled as digital input lines.
1	Analog Channel Mask	Each bit in the analog channel mask corresponds to one analog channel. The bits from LSB to MSB correspond to AIN0-AIN7 on the module. For example, if the analog channel mask is 0x06, AINI and AIN3 are enabled as analog input lines.
Var	Sampled Data Set	A sample set consisting of 1 sample for each enabled ADC and/or DIO channel. If any digital input lines are enabled, the first two bytes indicate the state of all enabled digital input lines. Each bit in these two bytes corresponds to one digital IO line, similar to the way each bit in the diglossia channel mask corresponds. Note: only the digital input line that are enabled in the detail channel mask have valid readings. Channels that are not enabled as digital input lines will return a 0 in the sampled data set. If no pins are configured as digital inputs, these 2 bytes will be omitted. Following the digital input data, if any, each enabled analog channel will return 2 bytes (10bits). The analog data is scaled such that 0 represents 0V, and 0x3FF=1.2V. The analog input lines cannot measure more than 1.2V. Information for each enabled analog channel is returned in order, starting with AIN0 and finishing with AIN4. Only enabled analog input channels will return data.

The AT firmware returns a carriage return delimited list containing the above-listed fields. The API firmware returns an AT command response API frame with the IO data included in the command data portion of the packet.

Table 4-013.

Example	Sample AT Response
0x01\r	[1 sample set]

Table 4-013.

Example	Sample AT Response
0x0C0C\r	[Digital Inputs: DIO 2, 3, 10, 11 low]
0x03\r	[Analog Inputs: A/D 0, 1]
0x0408\r	[Digital input states: DIO 3, 10 high, DIO 2, 11 low]
0x03D0\r	[Analog input ADIO 0= 0x3D0]
0x0124\r	[Analog input ADIO 1=0x120]

5. Network Commissioning and Diagnostics

Network commissioning is the process whereby devices in a mesh network are discovered and configured for operation. The XBee ZNet 2.5 modules include several features to support device discovery and configuration. In addition to configuring devices, a strategy must be developed to place devices to ensure reliable routes.

To accommodate these requirements, the XBee ZNet 2.5 modules include various features to aid in device placement, configuration, and network diagnostics.

5.1. Device Discovery

The node discovery command can be used to discover all modules that have joined a network. Issuing the ND command sends a broadcast node discovery command throughout the network. All devices that receive the command will send a response that includes the device's addressing information, node identifier string (see NI command), and other relevant information. This command is useful for generating a list of all module addresses in a network.

When a device receives the node discovery command, it waits a random time before sending its own response. The maximum time delay is set on the ND sender with the NT command. The ND originator includes its NT setting in the transmission to provide a delay window for all devices in the network. Large networks may need to increase NT to improve network discovery reliability. The default NT value is 0x3C (6 seconds).

5.2. Device Configuration

API devices can send configuration commands to remote modules to set or read the configuration settings of any device in the network.

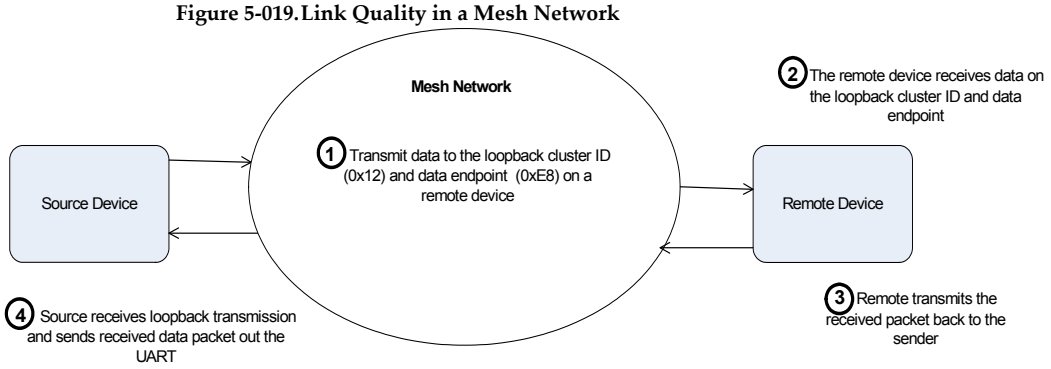
5.3. Device Placement

For a mesh network installation to be successful, the installer must be able to determine where to place individual XBee devices to establish reliable links throughout the mesh network.

5.3.1. Link Testing

A good way to measure the performance of a mesh network is to send unicast data through the network from one device to another to determine the success rate of many transmissions. To simplify link testing, the modules support a loopback cluster ID (0x12) on the data endpoint (0xE8). Any data sent to this cluster ID on the data endpoint will be transmitted back to the sender. This is shown in the figure below:

Figure 5-019. Link Quality in a Mesh Network


Mesh Network
1 Transmit data to the loopback cluster ID (0x12) and data endpoint (0xE8) on a remote device
2 The remote device receives data on the loopback cluster ID and data endpoint
3 Remote transmits the received packet back to the sender
4 Source receives loopback transmission and sends received data packet out the UART
Source Device
Remote Device

Demonstration of how the loopback cluster ID and data endpoint can be used to measure the link quality in a mesh network

The configuration steps to send data to the loopback cluster ID depend on the firmware type.

AT Firmware

To send data to the loopback cluster ID on the data endpoint of a remote device, set the ZA command to 1 and set the CI command value to 0x12. The SE and DE commands should be set to 0xE8 (default value). The DH and DL commands should be set to the address of the remote (0 for the coordinator, or the 64-bit address of the remote). After exiting command mode, any received serial characters will be transmitted to the remote device, and returned to the sender.

API Firmware

Send an Explicit Addressing ZigBee Command API frame (0x11) using 0x12 as the cluster ID and 0xE8 as the source and destination endpoint. Data packets received by the remote will be echoed back to the sender.

5.3.2. RSSI Indicators

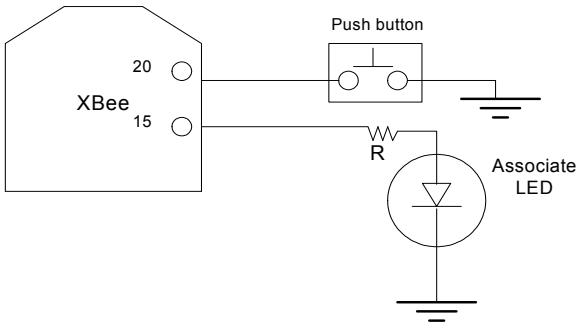
It is possible to measure the received signal strength on a device using the DB command. DB returns the RSSI value (measured in –dBm) of the last received packet. However, this number can be misleading. The DB value only indicates the received signal strength of the last hop. If a transmission spans multiple hops, the DB value provides no indication of the overall transmission path, or the quality of the worst link – it only indicates the quality of the last link and should be used sparingly.

The DB value can be determined in hardware using the RSSI/PWM module pin (pin 6). If the RSSI PWM functionality is enabled (P0 command), when the module receives data, the RSSI PWM is set to a value based on the RSSI of the received packet. (Again, this value only indicates the quality of the last hop.) This pin could potentially be connected to an LED to indicate if the link is stable or not.

5.4. Commissioning Pushbutton and Associate LED

The XBee modules support a set of commissioning and LED behaviors to aid in device deployment and commissioning. These include the commissioning push button definitions and associate LED behaviors. These features can be supported in hardware as shown below.

Figure 5-020. Commissioning Pushbutton and Associate LED Functionalities



A pushbutton and an LED can be connected to module pins 20 and 15 respectively to support the commissioning pushbutton and associate LED functionalities.

5.4.1. Commissioning Pushbutton

The commissioning pushbutton definitions provide a variety of simple functions to aid in deploying devices in a network. The commissioning button functionality on pin 20 is enabled by setting the D0 command to 1 (enabled by default)..

Table 5-014.

Button Presses	If module is joined to a network	If module is not joined to a network
1	• Wakes an end device for 30 seconds • Sends a node identification broadcast transmission	• Wakes an end device for 30 seconds • Blinks a numeric error code on the Associate pin indicating the cause of join failure (see section 6.4.2).
2	• Sends a broadcast transmission to enable joining on the coordinator and all devices in the network for 1 minute. (If joining is permanently enabled on a device (NJ = 0xFF), this action has no effect on that device.)	• N/A
4	• Causes the device to leave the PAN. • Issues ATRE to restore module parameters to default values, including ID and SC. • The device attempts to join a network based on its ID and SC settings.	• Issues ATRE to restore module parameters to default values, including ID and SC. • The device attempts to join a network based on its ID and SC settings.

Button presses may be simulated in software using the ATCB command. ATCB should be issued with a parameter set to the number of button presses to execute. (i.e. sending ATCB1 will execute the action(s) associated with a single button press.)

The node identification frame is similar to the node discovery response frame – it contains the device's address, node identifier string (NI command), and other relevant data. All API devices that receive the node identification frame send it out their Uart as an API Node Identification Indicator frame (0x95).

5.4.2. Associate LED

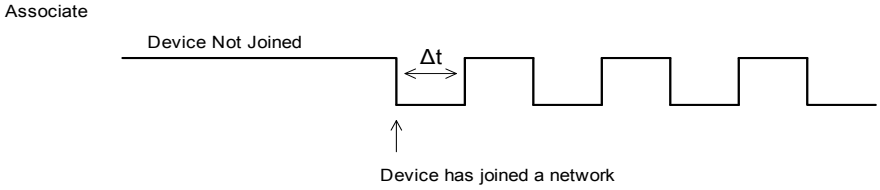
The Associate pin (pin 15) can provide indication of the device's network status and diagnostics information. To take advantage of these indications, an LED can be connected to the Associate pin as shown in the figure above. The Associate LED functionality is enabled by setting the D5 command to 1 (enabled by default). If enabled, the Associate pin is configured as an output and will behave as described in the following sections.

Joined Indication

The Associate pin indicates the network status of a device. If the module is not joined to a network, the Associate pin is set high. Once the module successfully joins a network, the Associate pin blinks at a regular time interval. This is shown in the following figure.

© 2008 Digi International, Inc.

Figure 5-021. Joined Status of a Device



The associate pin can indicate the joined status of a device . Once the device has joined a network, the associate pin toggles state at a regular interval (Δt). The time can be set by using the LT command.

The LT command defines the blink time of the Associate pin. If set to 0, the device uses the default blink time (500ms for coordinator, 250ms for routers and end devices).

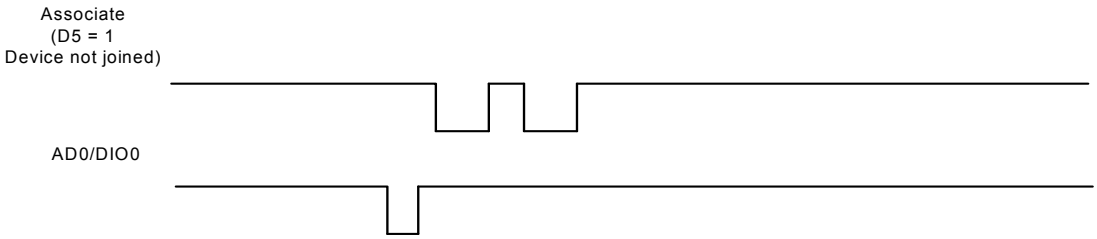
Diagnostics Support

The Associate pin works with the commissioning pushbutton to provide additional diagnostics behaviors to aid in deploying and testing a network. If the commissioning push button is pressed once, and the device has not joined a network, the Associate pin blinks a numeric error code to indicate the cause of join failure. The number of blinks is equal to (AI value – 0x20). For example, if AI=0x22, 2 blinks occur.

If the commissioning push button is pressed once, and the device has joined a network, the device transmits a broadcast node identification packet. If the Associate LED functionality is enabled (D5 command), a device that receive this transmission will blink its Associate pin rapidly for 1 second.

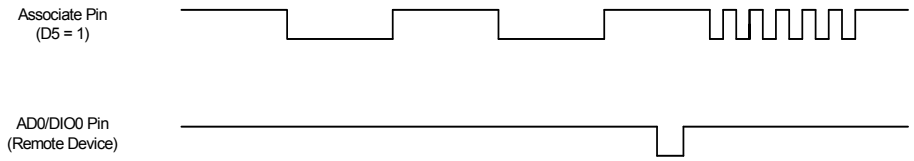
The following figures demonstrate these behaviors.

Figure 5-022. AI = 0x22



A single commissioning button press when the device has not joined a network tha causes the associate pin to blink to indicate the AI Code where: AI = # blinks + 0x2 In this example, AI = 0x22.

Figure 5-023. Broadcast Node Identification Transmission



A single button press on a remote device causes a broadcast node identification transmission to be sent. All devices that receive this transmission blink their associate pin rapidly for one second if the associate LED functionality is enabled. (D5 = 1)

6. API Operation

As an alternative to Transparent Operation, API (Application Programming Interface) Operations are available. API operation requires that communication with the module be done through a structured interface (data is communicated in frames in a defined order). The API specifies how commands, command responses and module status messages are sent and received from the module using a UART Data Frame.

Please note that Digi may add new API frames to future versions of firmware, so please build into your software interface the ability to filter out additional API frames with unknown API identifiers.

6.0.1. API Frame Specifications

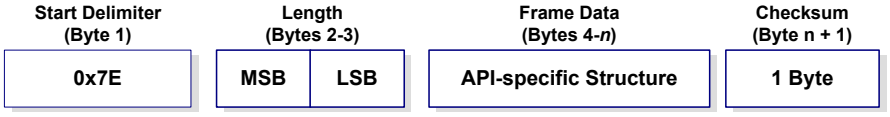
Two API modes are supported and both can be enabled using the AP (API Enable) command. Use the following AP parameter values to configure the module to operate in a particular mode:

- AP = 1: API Operation
- AP = 2: API Operation (with escaped characters)

API Operation (AP parameter = 1)

When this API mode is enabled (AP = 1), the UART data frame structure is defined as follows:

Figure 6-01. UART Data Frame Structure:



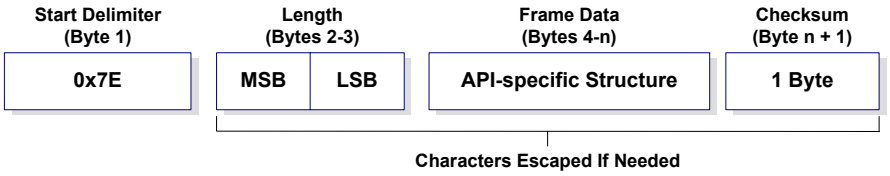
MSB = Most Significant Byte, LSB = Least Significant Byte

Any data received prior to the start delimiter is silently discarded. If the frame is not received correctly or if the checksum fails, the module will reply with a module status frame indicating the nature of the failure.

API Operation - with Escape Characters (AP parameter = 2)

When this API mode is enabled (AP = 2), the UART data frame structure is defined as follows:

Figure 6-02. UART Data Frame Structure - with escape control characters:



MSB = Most Significant Byte, LSB = Least Significant Byte

Escape characters. When sending or receiving a UART data frame, specific data values must be escaped (flagged) so they do not interfere with the data frame sequencing. To escape an interfering data byte, insert 0x7D and follow it with the byte to be escaped XOR'd with 0x20.

Data bytes that need to be escaped:

- 0x7E – Frame Delimiter
- 0x7D – Escape
- 0x11 – XON
- 0x13 – XOFF

Example - Raw UART Data Frame (before escaping interfering bytes):
0x7E 0x00 0x02 0x23 0x11 0xCB

0x11 needs to be escaped which results in the following frame:
0x7E 0x00 0x02 0x23 0x7D 0x31 0xCB

Note: In the above example, the length of the raw data (excluding the checksum) is 0x0002 and the checksum of the non-escaped data (excluding frame delimiter and length) is calculated as: 0xFF - (0x23 + 0x11) = (0xFF - 0x34) = 0xCB.

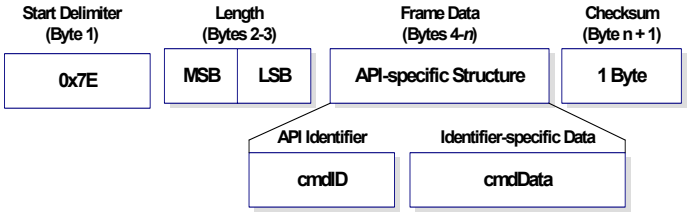
Length

The length field has two-byte value that specifies the number of bytes that will be contained in the frame data field. It does not include the checksum field.

Frame Data

Frame data of the UART data frame forms an API-specific structure as follows:

Figure 6-03. UART Data Frame & API-specific Structure:



The cmdID frame (API-identifier) indicates which API messages will be contained in the cmdData frame (Identifier-specific data). Note that multi-byte values are sent big endian.The XBee ZNet 2.5 modules support the following API frames:

Table 6-015. API Frame Names and Values

API Frame Names	Values
Modem Status	0x8A
AT Command	0x08
AT Command - Queue Parameter Value	0x09
AT Command Response	0x88
Remote Command Request	0x17
Remote Command Response	0x97
ZigBee Transmit Request	0x10
Explicit Addressing ZigBee Command Frame	0x11
ZigBee Transmit Status	0x8B
ZigBee Receive Packet (AO=0)	0x90
ZigBee Explicit Rx Indicator (AO=1)	0x91
ZigBee IO Data Sample Rx Indicator	(0x92)
XBee Sensor Read Indicator (AO=0)	0x94
Node Identification Indicator (AO=0)	0x95

Checksum

To test data integrity, a checksum is calculated and verified on non-escaped data.

To calculate: Not including frame delimiters and length, add all bytes keeping only the lowest 8 bits of the result and subtract the result from 0xFF.

To verify: Add all bytes (include checksum, but not the delimiter and length). If the checksum is correct, the sum will equal 0xFF.

6.0.2. API Examples

Example: Create an API AT command frame to configure an XBee to allow joining (set NJ to 0xFF). The frame should look like:

0x7E 0x00 0x05 0x08 0x01 0x4E 0x4A 0xFF 5F

Where 0x0005 = length

0x08 = AT Command API frame type

0x01 = Frame ID (set to non-zero value)

0x4E4A = AT Command ('NJ')

0xFF = value to set command to

0x5F = Checksum

The checksum is calculated as [0xFF - (0x08 + 0x01 + 0x4E + 0x4A + 0xFF)]

Example: Send a transmission to a module with destination address 0x0013A200 40014011, payload "TxData1B". If escaping is disabled, (AP=1), the frame should look like:

0x7E 0x00 0x16 0x10 0x01 0x00 0x13 0xA2 0x00 0x40 0x0A 0x01 0x27 0xFF

0xFE 0x00 0x00 0x54 0x78 0x44 0x61 0x74 0x61 0x30 0x41 0x13

Where 0x16 = length (22 bytes excluding checksum)

0x10 = ZigBee Transmit Request API frame type

0x01 = Frame ID (set to non-zero value)

0x0013A200400A0127 = 64-bit Destination Address

0xFFFE = 16-bit Destination Address

0x00 = Broadcast radius

0x00 = Options

0x5478446174613041 = Data payload ("TxData0A")

0x64 = Checksum

If escaping is enabled (AP=2), the frame should look like:

0x7E 0x00 0x16 0x10 0x01 0x00 0x7D 0x33 0xA2 0x00 0x40 0x0A 0x01 0x27

0xFF 0xFE 0x00 0x00 0x54 0x78 0x44 0x61 0x74 0x61 0x30 0x41 0x7D 0x33

The checksum is calculated (on all non-escaped bytes) as [0xFF - (sum of all bytes from API frame type through data payload)].

Example: Send a transmission to the coordinator without specifying the coordinator's 64-bit address. The API transmit request frame should look like:

0x7E 0x00 0x16 0x10 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0xFE 0x00 0x00 0x54 0x78 032 0x43 0x6F 0x6F 0x72 0x64 0xFC

Where 0x16 = length (22 bytes excluding checksum)

0x10 = ZigBee Transmit Request API frame type

0x01 = Frame ID (set to non-zero value)

0x0000000000000000 = Coordinator's address (can be replaced with coordinator's actual 64-bit address if known

0xFFFE = 16-bit Destination Address

0x00 = Broadcast radius

0x00 = Options

0x547832436F6F7264 = Data payload ("Tx2Coord")

0xFC = Checksum

Example: Send an ND command to discover the devices in the PAN. The frame should look like:

0x7E 0x00 0x04 0x08 0x01 0x4E 0x44 0x64

Where 0x0004 = length

0x08 = AT Command API frame type

0x01 = Frame ID (set to non-zero value)

0x4E44 = AT command ('ND')

0x64 = Checksum

The checksum is calculated as [0xFF - (0x08 + 0x01 + 0x4E + 0x44)]

6.0.3. Supporting the API

Applications that support the API should make provisions to deal with new API frames that may be introduced in future releases. For example, a section of code on a host microprocessor that handles received serial API frames (sent out the module's DOUT pin) might look like this:

```
void XBee_HandleRxAPIFrame(_apiFrameUnion *papiFrame){
  switch(papiFrame->api_id){
    case RX_RF_DATA_FRAME:
      //process received RF data frame
      break;

    case RX_IO_SAMPLE_FRAME:
      //process IO sample frame
      break;

    case NODE_IDENTIFICATION_FRAME:
      //process node identification frame
      break;

    default:
      //Discard any other API frame types that are not being used
      break;
  }
}
```

6.0.4. API Frames

The following sections illustrate the types of frames encountered while using the API.

Note: "The following information is missing or incorrect in the API graphics:

* Transmit option 0x08 is not supported

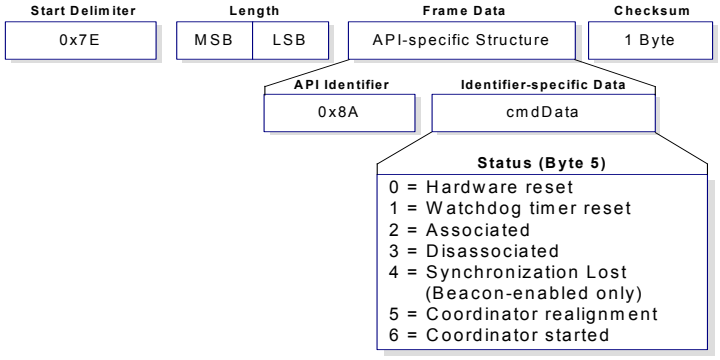
The following information is not shown in the API graphics:

- The 64-bit destination address can be set to 0 as an alternate address for the coordinator.
- The maximum broadcast radius is 32.
- The NO command can be used to include the DD parameter in the Node Identification Indicator (0x95).

Modem Status

API Identifier Value: (0x8A)
RF module status messages are sent from the module in response to specific conditions.

Figure 6-04. Modem Status Frames



AT Command

API Identifier Value: 0x08
Allows for module parameter registers to be queried or set.

Figure 6-5. AT Command Frames

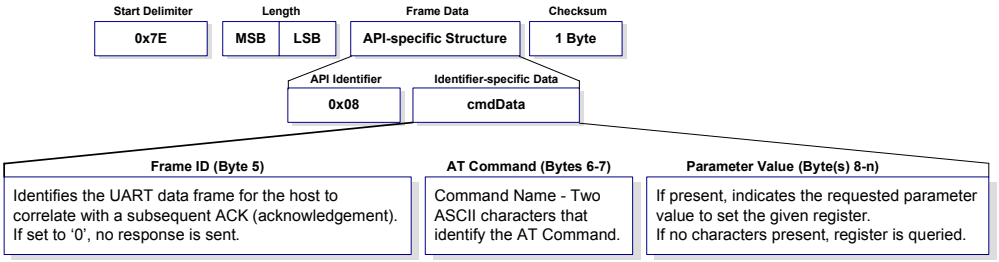
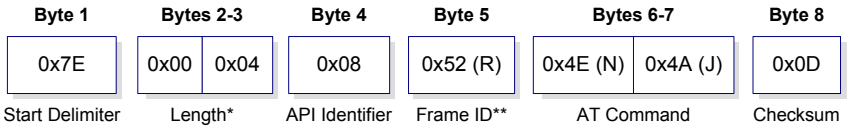
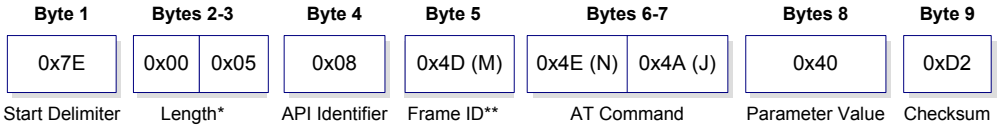

Start Delimiter
0x7E
Length
MSB
LSB
Frame Data
API-specific Structure
Checksum
1 Byte
API Identifier
0x08
Identifier-specific Data
cmdData
Frame ID (Byte 5)
Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to '0', no response is sent.
AT Command (Bytes 6-7)
Command Name - Two ASCII characters that identify the AT Command.
Parameter Value (Byte(s) 8-n)
If present, indicates the requested parameter value to set the given register. If no characters present, register is queried.

Figure 6-6. Example: API frames when reading the NJ parameter value of the module.



** Length [Bytes] = API Identifier + Frame ID + AT Command*
*** "R" value was arbitrarily selected.*

Figure 6-7. Example: API frames when modifying the NJ parameter value of the module.



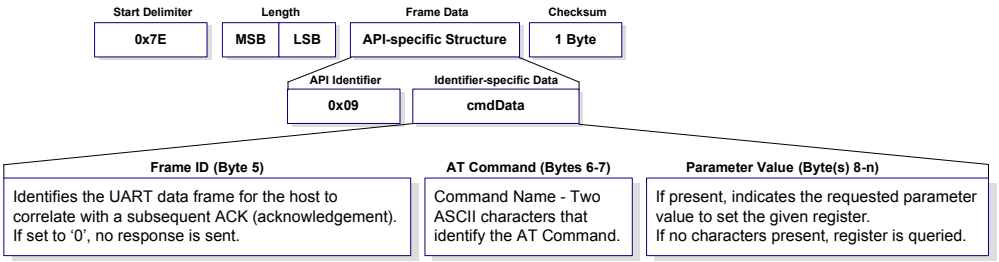
** Length [Bytes] = API Identifier + Frame ID + AT Command + Parameter Value*
*** "M" value was arbitrarily selected.*

A string parameter used with the NI (Node Identifier), ND (Node Discover) and DH (Destination Address High) command is terminated with a 0x00 character.

AT Command - Queue Parameter Value

API Identifier Value: 0x09
This API type allows module parameters to be queried or set. In contrast to the "AT Command" API type, new parameter values are queued and not applied until either the "AT Command" (0x08) API type or the AC (Apply Changes) command is issued. Register queries (reading parameter values) are returned immediately.

Figure 6-8. AT Command Frames
(Note that frames are identical to the "AT Command" API type except for the API identifier.)

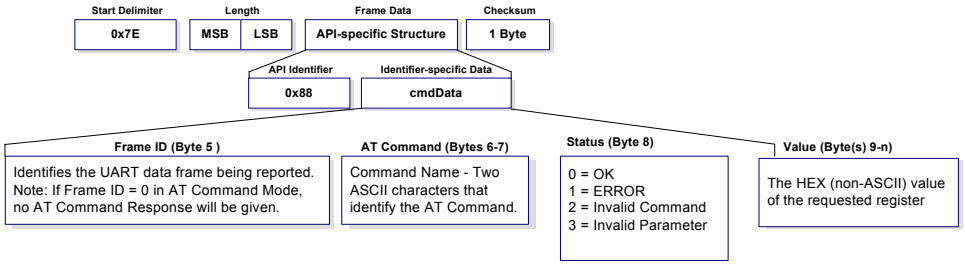

Start Delimiter
0x7E
Length
MSB
LSB
Frame Data
API-specific Structure
Checksum
1 Byte
API Identifier
0x09
Identifier-specific Data
cmdData
Frame ID (Byte 5)
Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to '0', no response is sent.
AT Command (Bytes 6-7)
Command Name - Two ASCII characters that identify the AT Command.
Parameter Value (Byte(s) 8-n)
If present, indicates the requested parameter value to set the given register. If no characters present, register is queried.

AT Command Response

API Identifier Value: 0x88
Response to previous command.

In response to an AT Command message, the module will send an AT Command Response message. Some commands will send back multiple frames (for example, the ND (Node Discover) command).

Figure 6-9. AT Command Response Frames.

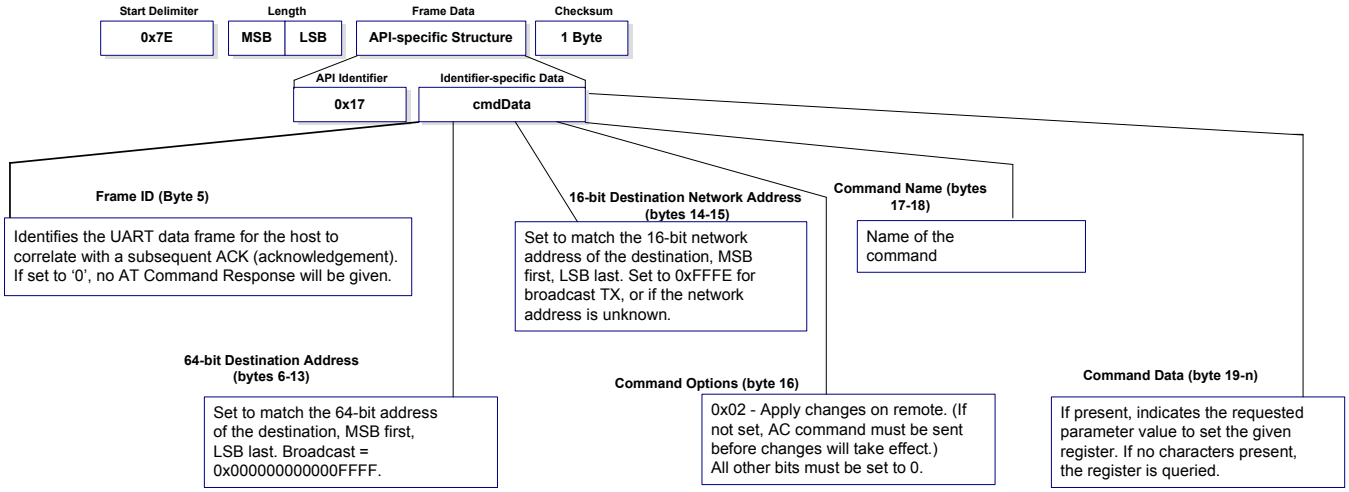


Remote AT Command Request

API Identifier Value: 0x17

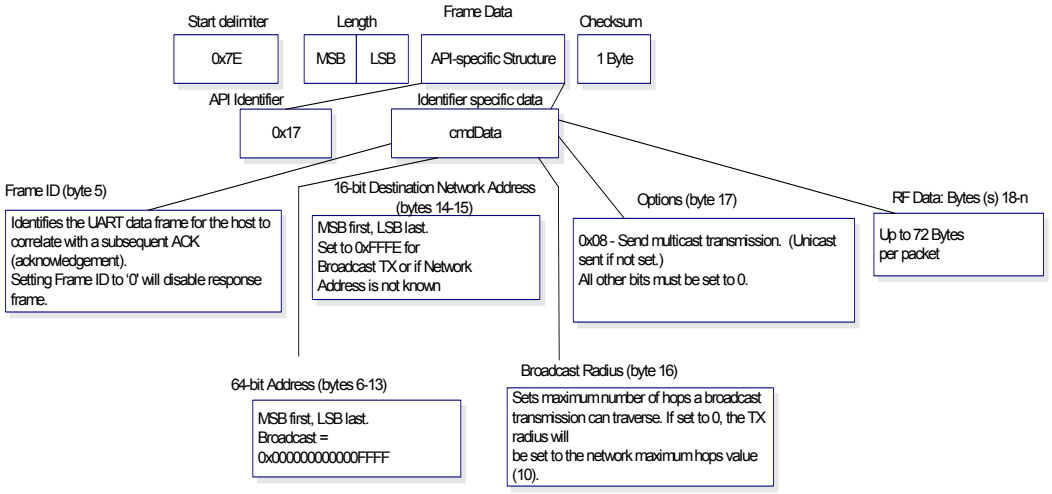
Allows for module parameter registers on a remote device to be queried or set

Figure 6-10. Remote AT Command Request


Start Delimiter
0x7E
Length
MSB
LSB
Frame Data
API-specific Structure
Checksum
1 Byte
API Identifier
0x17
Identifier-specific Data
cmdData
Frame ID (Byte 5)
Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). If set to '0', no AT Command Response will be given.
16-bit Destination Network Address (bytes 14-15)
Set to match the 16-bit network address of the destination, MSB first, LSB last. Set to 0xFFFE for broadcast TX, or if the network address is unknown.
Command Name (bytes 17-18)
Name of the command
64-bit Destination Address (bytes 6-13)
Set to match the 64-bit address of the destination, MSB first, LSB last. Broadcast = 0x000000000000FFFF.
Command Options (byte 16)
0x02 - Apply changes on remote. (If not set, AC command must be sent before changes will take effect.) All other bits must be set to 0.
Command Data (byte 19-n)
If present, indicates the requested parameter value to set the given register. If no characters present, the register is queried.

© 2008 Digi International, Inc.

Figure 6-11.

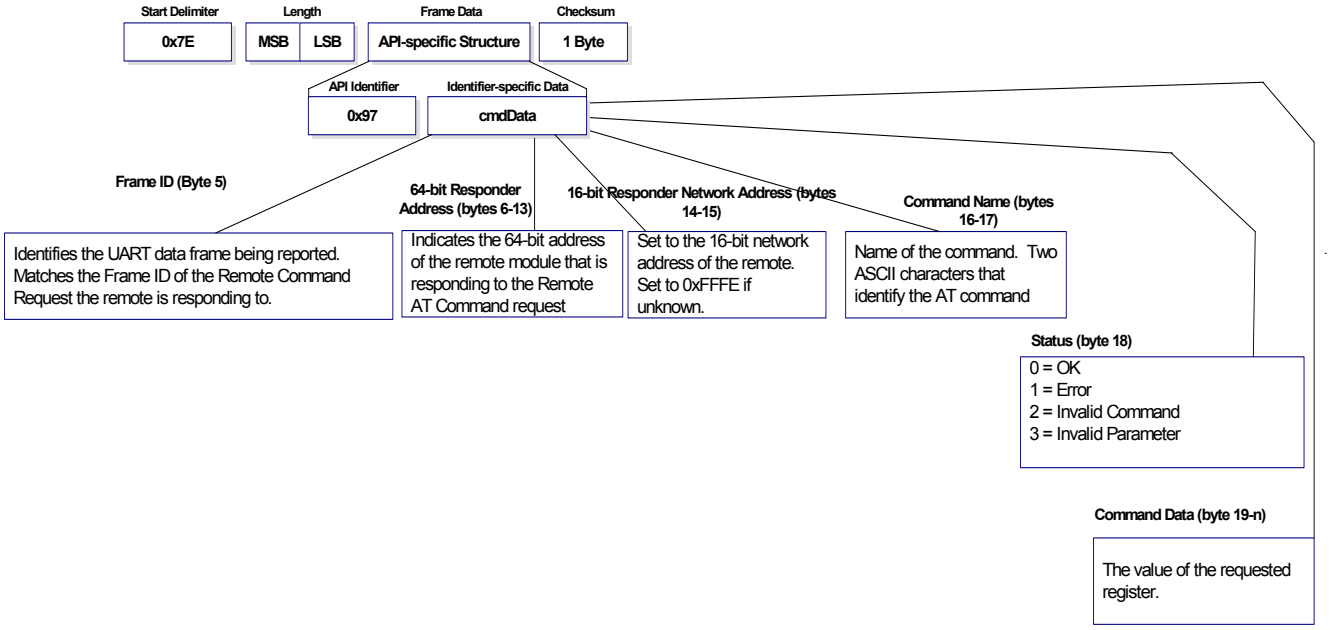

Start delimiter
0x7E
Length
MSB
LSB
Frame Data
API-specific Structure
Checksum
1 Byte
API Identifier
0x17
Identifier specific data
cmdData
Frame ID (byte 5)
Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement).
Setting Frame ID to '0' will disable response frame.
16-bit Destination Network Address (bytes 14-15)
MSB first, LSB last.
Set to 0xFFFE for Broadcast TX or if Network Address is not known
Options (byte 17)
0x08 - Send multicast transmission. (Unicast sent if not set.)
All other bits must be set to 0.
RF Data: Bytes (s) 18-n
Up to 72 Bytes per packet
64-bit Address (bytes 6-13)
MSB first, LSB last.
Broadcast = 0x000000000000FFFF
Broadcast Radius (byte 16)
Sets maximum number of hops a broadcast transmission can traverse. If set to 0, the TX radius will
be set to the network maximum hops value (10).

Remote Command Response

API Identifier Value: 0x97

If a module receives a remote command response RF data frame in response to a Remote AT Command Request, the module will send a Remote AT Command Response message out the UART. Some commands may send back multiple frames--for example, Node Discover (ND) command.

Figure 6-12. Remote AT Command Response.

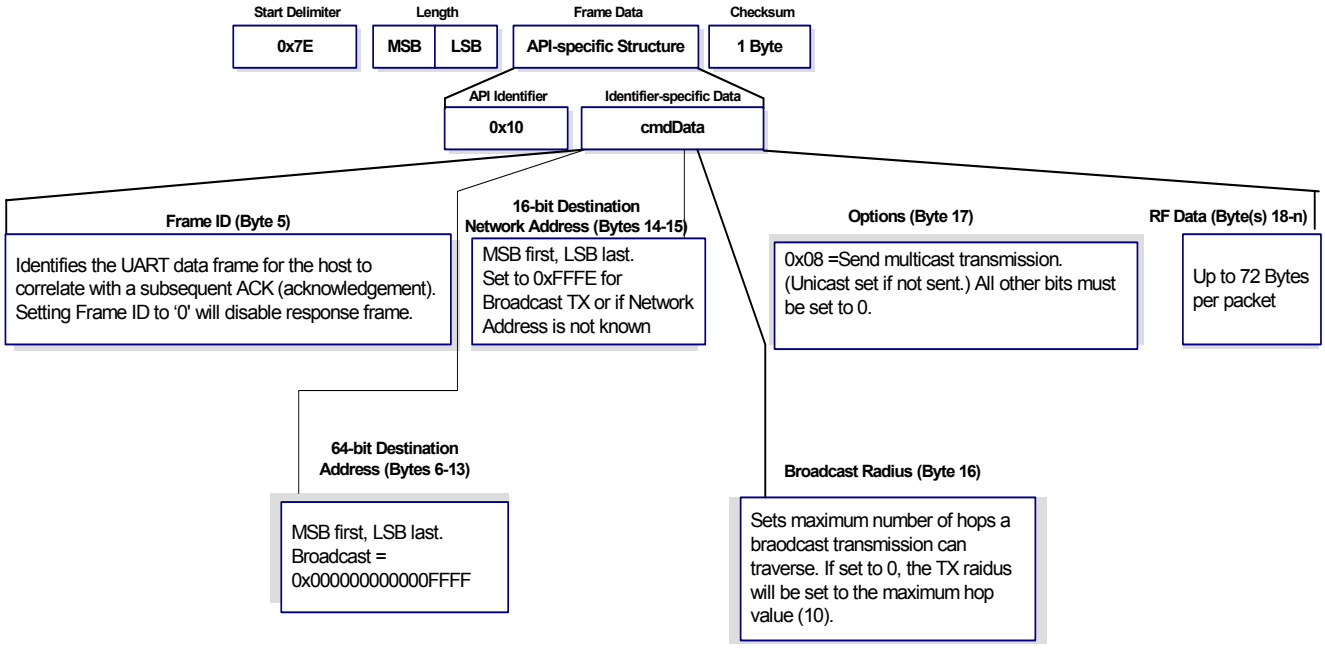

Start Delimiter
0x7E
Length
MSB
LSB
Frame Data
API-specific Structure
Checksum
1 Byte
API Identifier
0x97
Identifier-specific Data
cmdData
Frame ID (Byte 5)
Identifies the UART data frame being reported. Matches the Frame ID of the Remote Command Request the remote is responding to.
64-bit Responder Address (bytes 6-13)
Indicates the 64-bit address of the remote module that is responding to the Remote AT Command request
16-bit Responder Network Address (bytes 14-15)
Set to the 16-bit network address of the remote. Set to 0xFFFE if unknown.
Command Name (bytes 16-17)
Name of the command. Two ASCII characters that identify the AT command
Status (byte 18)
0 = OK
1 = Error
2 = Invalid Command
3 = Invalid Parameter
Command Data (byte 19-n)
The value of the requested register.

ZigBee Transmit Request

API Identifier Value: 0x10

A TX Request message will cause the module to send RF Data as an RF Packet.TX Packet Frames

Figure 6-13. ZigBee Transmit Request.

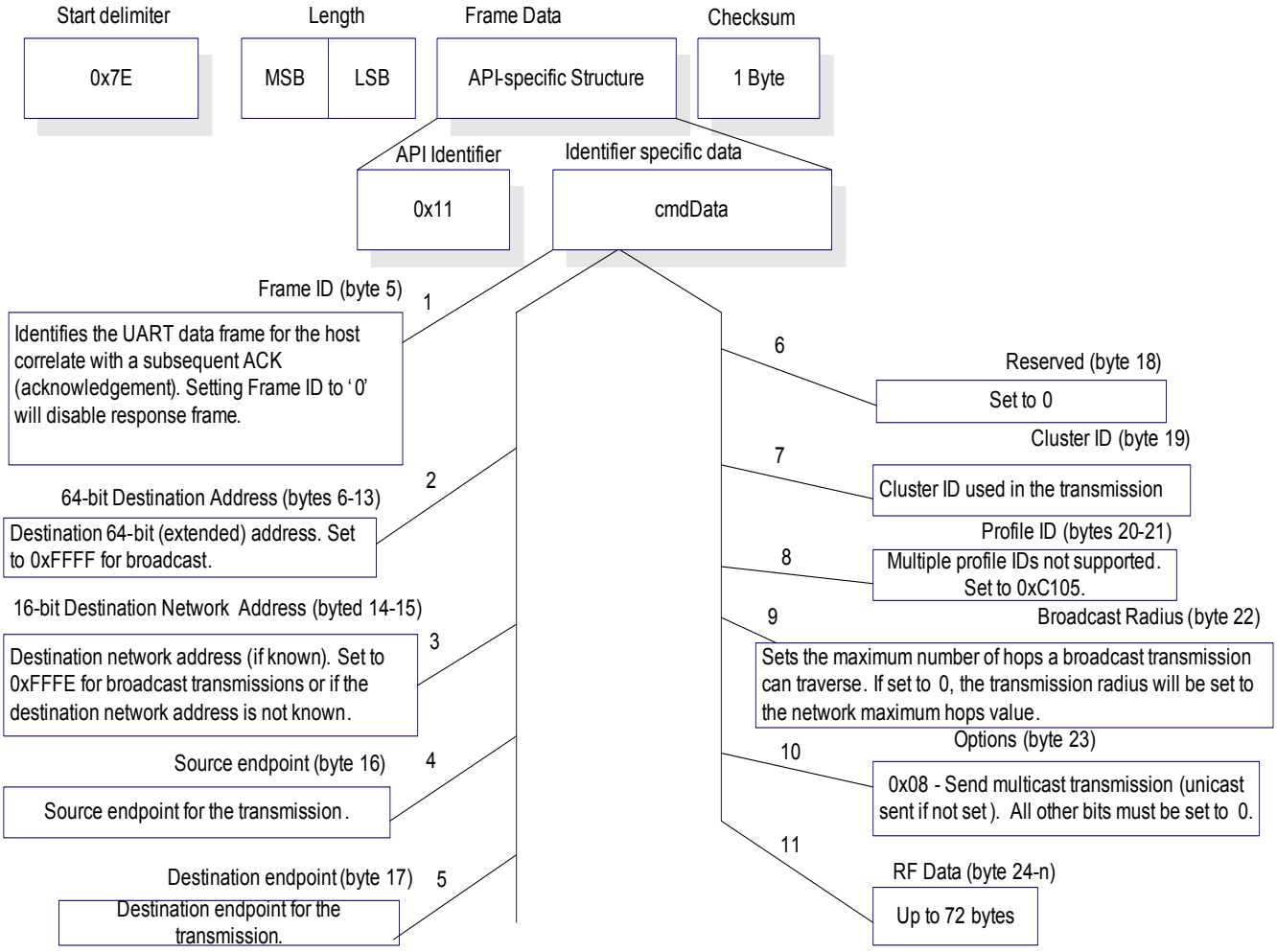

Start Delimiter
0x7E
Length
MSB
LSB
Frame Data
API-specific Structure
Checksum
1 Byte
API Identifier
0x10
Identifier-specific Data
cmdData
Frame ID (Byte 5)
Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgement). Setting Frame ID to '0' will disable response frame.
16-bit Destination Network Address (Bytes 14-15)
MSB first, LSB last. Set to 0xFFFE for Broadcast TX or if Network Address is not known
Options (Byte 17)
0x08 =Send multicast transmission. (Unicast set if not sent.) All other bits must be set to 0.
RF Data (Byte(s) 18-n)
Up to 72 Bytes per packet
64-bit Destination Address (Bytes 6-13)
MSB first, LSB last. Broadcast = 0x000000000000FFFF
Broadcast Radius (Byte 16)
Sets maximum number of hops a braodcast transmission can traverse. If set to 0, the TX raidus will be set to the maximum hop value (10).

Explicit Addressing ZigBee Command Frame

API Identifier Value: 0x11

Allows ZigBee application layer fields (endpoint and cluster ID) to be specified for a data transmission.

Figure 6-14. Explicit Addressing ZigBee Command Frame.

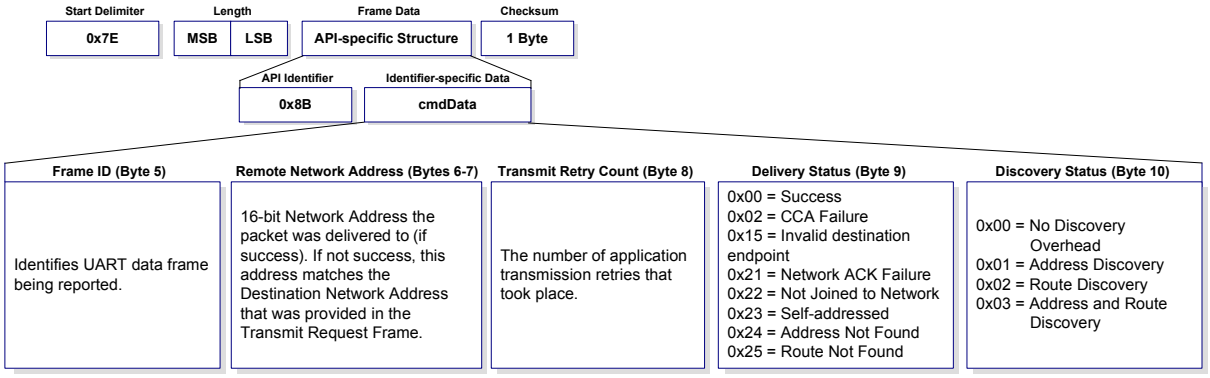

Start delimiter
0x7E
Length
MSB
LSB
Frame Data
API-specific Structure
Checksum
1 Byte
API Identifier
0x11
Identifier specific data
cmdData
Frame ID (byte 5)
1
Identifies the UART data frame for the host correlate with a subsequent ACK (acknowledgement). Setting Frame ID to '0' will disable response frame.
64-bit Destination Address (bytes 6-13)
2
Destination 64-bit (extended) address. Set to 0xFFFF for broadcast.
16-bit Destination Network Address (byted 14-15)
3
Destination network address (if known). Set to 0xFFFE for broadcast transmissions or if the destination network address is not known.
Source endpoint (byte 16)
4
Source endpoint for the transmission.
Destination endpoint (byte 17)
5
Destination endpoint for the transmission.
6
Reserved (byte 18)
Set to 0
7
Cluster ID (byte 19)
Cluster ID used in the transmission
8
Profile ID (bytes 20-21)
Multiple profile IDs not supported. Set to 0xC105.
9
Broadcast Radius (byte 22)
Sets the maximum number of hops a broadcast transmission can traverse. If set to 0, the transmission radius will be set to the network maximum hops value.
10
Options (byte 23)
0x08 - Send multicast transmission (unicast sent if not set). All other bits must be set to 0.
11
RF Data (byte 24-n)
Up to 72 bytes

ZigBee Transmit Status

API Identifier Value: 0x8B

When a TX Request is completed, the module sends a TX Status message. This message will indicate if the packet was transmitted successfully or if there was a failure.

Figure 6-15. TX Status Frames

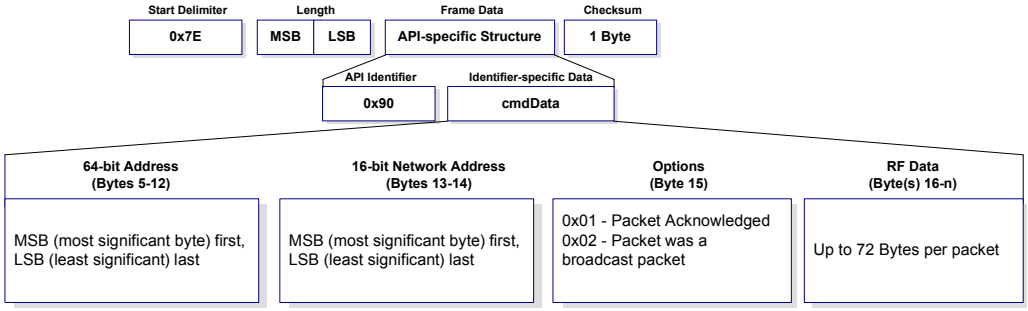

Start Delimiter
0x7E
Length
MSB
LSB
Frame Data
API-specific Structure
Checksum
1 Byte
API Identifier
0x8B
Identifier-specific Data
cmdData
Frame ID (Byte 5)
Identifies UART data frame being reported.
Remote Network Address (Bytes 6-7)
16-bit Network Address the packet was delivered to (if success). If not success, this address matches the Destination Network Address that was provided in the Transmit Request Frame.
Transmit Retry Count (Byte 8)
The number of application transmission retries that took place.
Delivery Status (Byte 9)
0x00 = Success
0x02 = CCA Failure
0x15 = Invalid destination endpoint
0x21 = Network ACK Failure
0x22 = Not Joined to Network
0x23 = Self-addressed
0x24 = Address Not Found
0x25 = Route Not Found
Discovery Status (Byte 10)
0x00 = No Discovery Overhead
0x01 = Address Discovery
0x02 = Route Discovery
0x03 = Address and Route Discovery

ZigBee Receive Packet

API Identifier Value: (0x90)

When the module receives an RF packet, it is sent out the UART using this message type.

Figure 6-16. RX Packet Frames

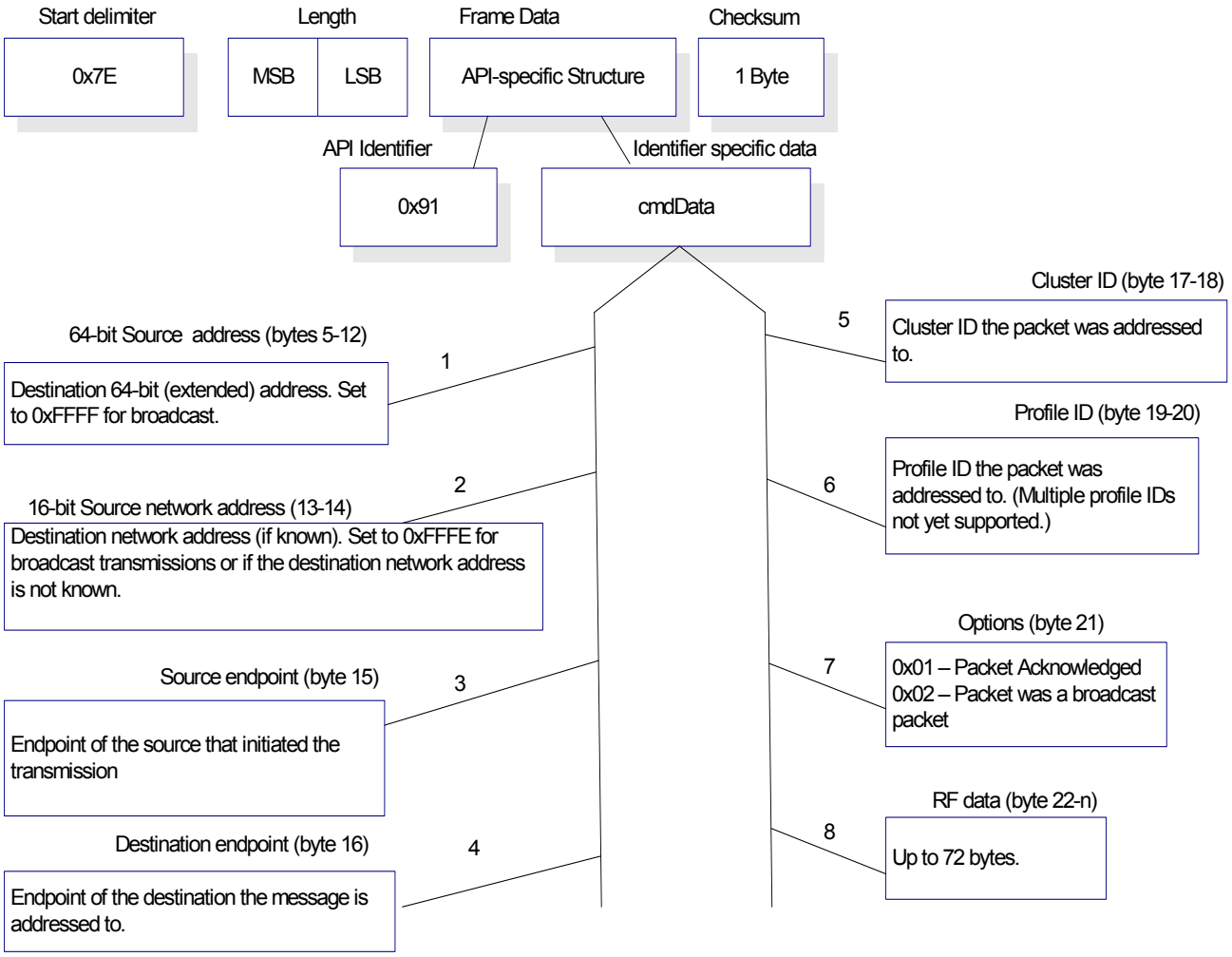


ZigBee Explicit Rx Indicator

API Identifier Value:0x91

When the modem receives a ZigBee RF packet it is sent out the UART using this message type (when AO=1).

Figure 6-17. ZigBee Explicit Rx Indicators

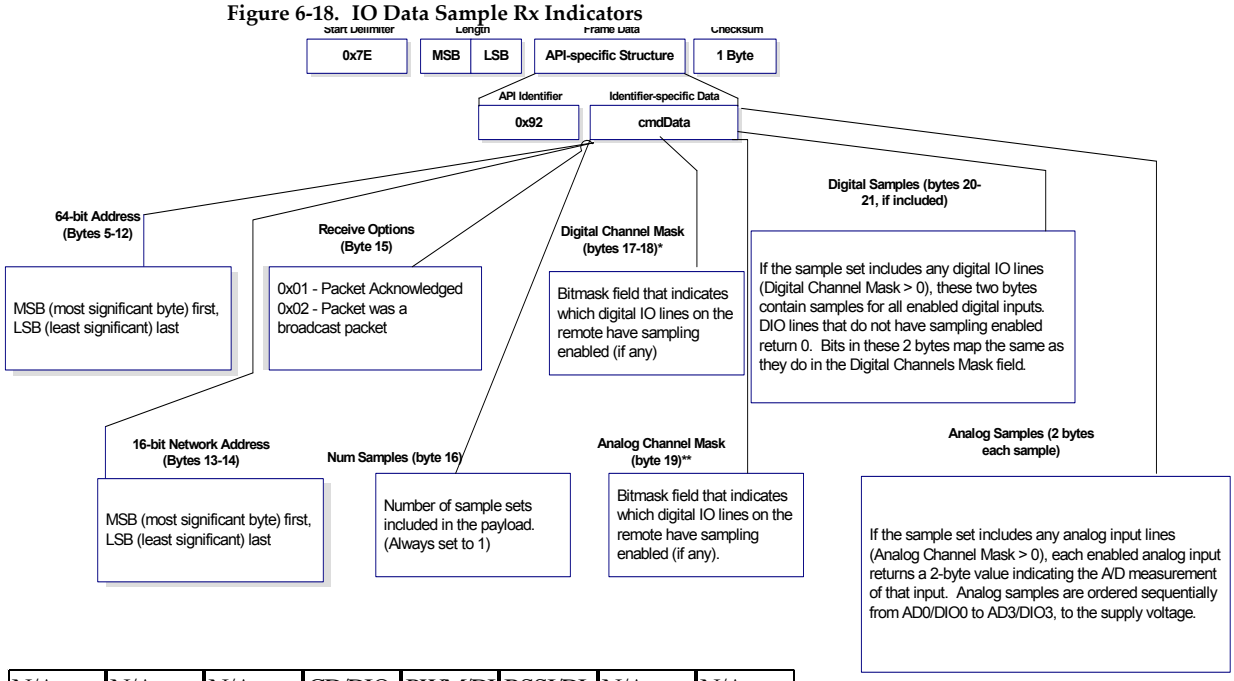


ZigBee IO Data Sample Rx Indicator

API Identifier Value: 0x92

When the module receives an IO sample frame from a remote device, it sends the sample out the UART using this frame type.

Figure 6-18. IO Data Sample Rx Indicators


Start Delimiter
Length
Frame Data
Checksum
0x7E
MSB
LSB
API-specific Structure
1 Byte
API Identifier
Identifier-specific Data
0x92
cmdData
64-bit Address (Bytes 5-12)
MSB (most significant byte) first, LSB (least significant) last
Receive Options (Byte 15)
0x01 - Packet Acknowledged
0x02 - Packet was a broadcast packet
Digital Channel Mask (bytes 17-18)*
Bitmask field that indicates which digital IO lines on the remote have sampling enabled (if any)
Digital Samples (bytes 20-21, if included)
If the sample set includes any digital IO lines (Digital Channel Mask > 0), these two bytes contain samples for all enabled digital inputs. DIO lines that do not have sampling enabled return 0. Bits in these 2 bytes map the same as they do in the Digital Channels Mask field.
16-bit Network Address (Bytes 13-14)
MSB (most significant byte) first, LSB (least significant) last
Num Samples (byte 16)
Number of sample sets included in the payload. (Always set to 1)
Analog Channel Mask (byte 19)**
Bitmask field that indicates which digital IO lines on the remote have sampling enabled (if any).
Analog Samples (2 bytes each sample)
If the sample set includes any analog input lines (Analog Channel Mask > 0), each enabled analog input returns a 2-byte value indicating the A/D measurement of that input. Analog samples are ordered sequentially from AD0/DIO0 to AD3/DIO3, to the supply voltage.

*

N/A	N/A	N/A	CD/DIO 12	PWM/DI O11	RSSI/DI O10	N/A	N/A
CTS/DI O7	RTS/DI O6	ASSOC/ DIO5	DIO4	AD3/DI O3	AD2/DI O2	AD1/DI O1	AD0/DI O0

**

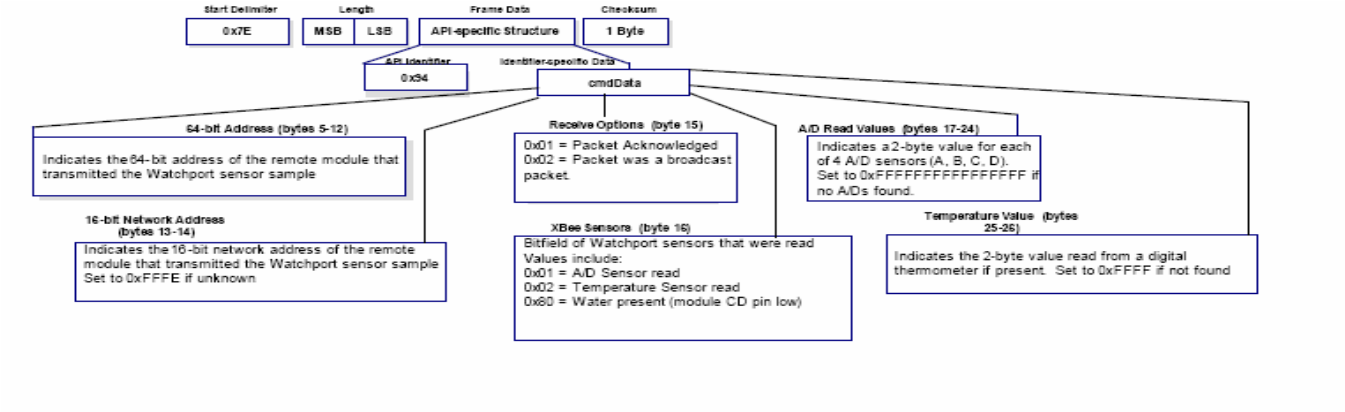
Supply Voltage	N/A	N/A	N/A	AD3	AD2	AD1	AD0

XBee Sensor Read Indicator

API Identifier Value: 0x94

When the module receives a sensor sample, it is sent out the UART using this message type (when AO=0).

Figure 6-19. XBee Sensor Read Indicator (0x94)

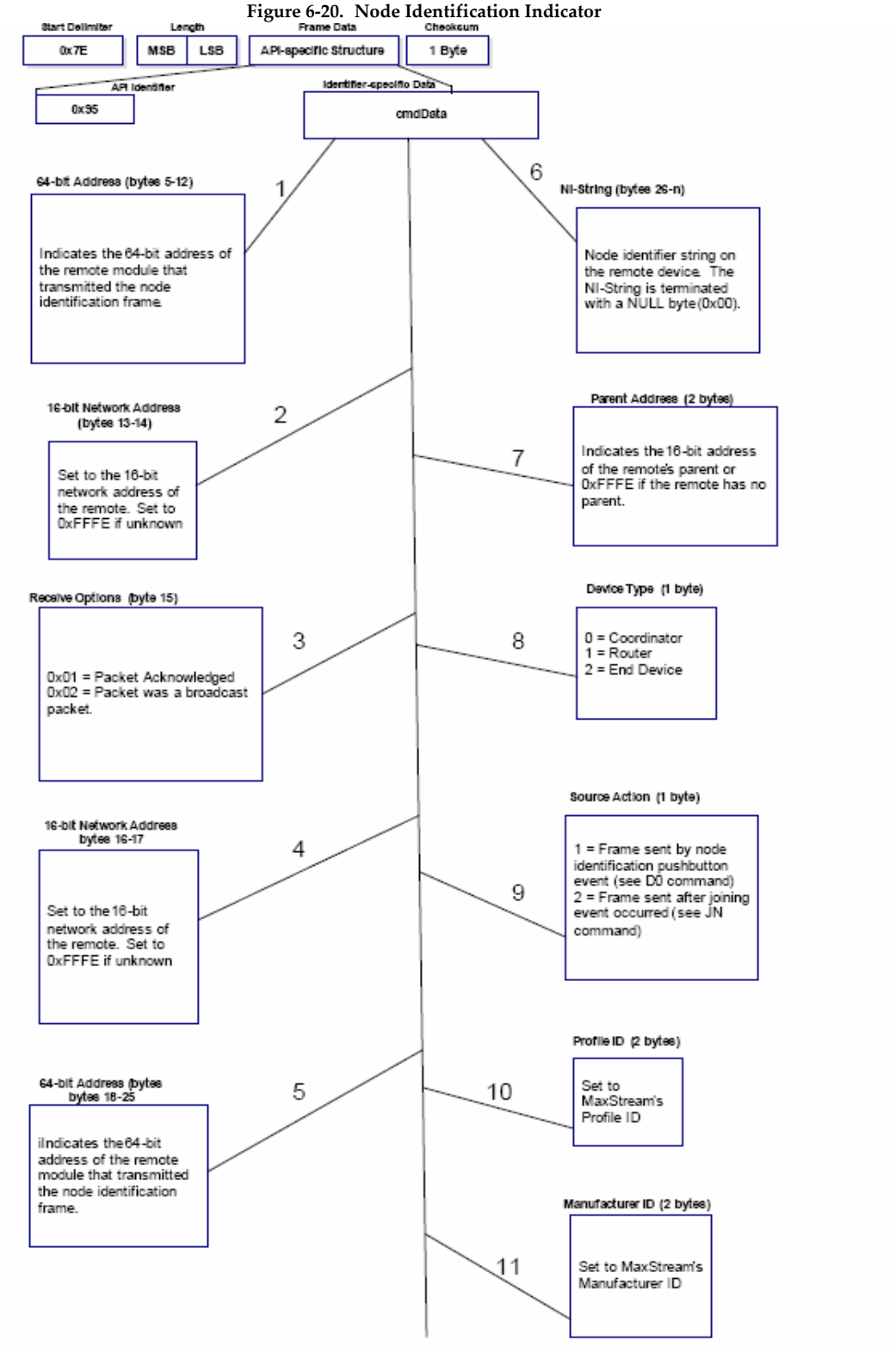

Start Delimiter
0x7E
Length
MSB
LSB
Frame Data
API-specific Structure
Checksum
1 Byte
API Identifier
0x94
Identifier-specific Data
cmdData
64-bit Address (bytes 5-12)
Indicates the 64-bit address of the remote module that transmitted the Watchport sensor sample
16-bit Network Address (bytes 13-14)
Indicates the 16-bit network address of the remote module that transmitted the Watchport sensor sample Set to 0xFFFE if unknown
Receive Options (byte 15)
0x01 = Packet Acknowledged
0x02 = Packet was a broadcast packet
XBee Sensors (byte 16)
Bitfield of Watchport sensors that were read Values include:
0x01 = A/D Sensor read
0x02 = Temperature Sensor read
0x80 = Water present (module CD pin low)
A/D Read Values (bytes 17-24)
Indicates a 2-byte value for each of 4 A/D sensors (A, B, C, D). Set to 0xFFFFFFFFFFFFFFFF if no A/Ds found.
Temperature Value (bytes 25-26)
Indicates the 2-byte value read from a digital thermometer if present. Set to 0xFFFF if not found

Node Identification Indicator

API Identifier Value: 0x95

This frame is received on the coordinator when a module transmits a node identification message to identify itself to the coordinator (when AO=0). The data portion of this frame is similar to a Node Discovery response frame (see ND command).

Figure 6-20. Node Identification Indicator


Start Delimiter
0x7E
Length
MSB
LSB
Frame Data
API-specific Structure
Checksum
1 Byte
API Identifier
0x95
Identifier-specific Data
cmdData
64-bit Address (bytes 5-12)
1
Indicates the 64-bit address of the remote module that transmitted the node identification frame.
6
NI-String (bytes 26-n)
Node identifier string on the remote device. The NI-String is terminated with a NULL byte(0x00).
16-bit Network Address (bytes 13-14)
2
Set to the 16-bit network address of the remote. Set to 0xFFFE if unknown
7
Parent Address (2 bytes)
Indicates the 16-bit address of the remote's parent or 0xFFFE if the remote has no parent.
Receive Options (byte 15)
3
0x01 = Packet Acknowledged
0x02 = Packet was a broadcast packet.
8
Device Type (1 byte)
0 = Coordinator
1 = Router
2 = End Device
16-bit Network Address bytes 16-17
4
Set to the 16-bit network address of the remote. Set to 0xFFFE if unknown
9
Source Action (1 byte)
1 = Frame sent by node identification pushbutton event (see D0 command)
2 = Frame sent after joining event occurred (see JN command)
64-bit Address (bytes bytes 18-25
5
iIndicates the 64-bit address of the remote module that transmitted the node identification frame.
10
Profile ID (2 bytes)
Set to MaxStream's Profile ID
11
Manufacturer ID (2 bytes)
Set to MaxStream's Manufacturer ID

7. XBee ZNet 2.5 Command Reference Tables

Special

Table 7-01. Special Commands

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
WR	**Write**. Write parameter values to non-volatile memory so that parameter modifications persist through subsequent resets. Note: Once WR is issued, no additional characters should be sent to the module until after the "OK\r" response is received. The WR command should be used sparingly. The EM250 supports a limited number of write cycles.“	CRE	--	--
WB	**Write Binding Table:** Writes the current binding table to non-volative memory.	CRE	--	--
RE	**Restore Defaults**. Restore module parameters to factory defaults. RE command does not reset the ID parameter.	CRE	--	--
FR	**Software Reset**. Reset module. Responds immediately with an “OK” then performs a reset ~2 seconds later. Use of the FR command will cause a network layer restart on the node if SC or ID were modified since the last reset.	CRE	--	--
NR	**Network Reset**. Reset network layer parameters on one or more modules within a PAN. Responds immediately with an “OK” then causes a network restart. All network configuration and routing information is consequently lost. *If NR = 0*: Resets network layer parameters on the node issuing the command. *If NR = 1*: Sends broadcast transmission to reset network layer parameters on all nodes in the PAN.	CRE	0 - 1	--

Node types that support the command: C = Coordinator, R = Router, E = End Device

Addressing

Table 7-02. Addressing Commands)

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
DH[2]	**Destination Address High**. Set/Get the upper 32 bits of the 64-bit destination address. When combined with DL, it defines the destination address used for transmission. 0x000000000000FFFF is the broadcast address for the PAN. DH is not supported in API Mode. 0x0000000000000000 is the Coordinator’s 16-bit network address.	CRE	0 - 0xFFFFFFFF	0
DL[2]	**Destination Address Low**. Set/Get the lower 32 bits of the 64-bit destination address. When combined with DH, DL defines the destination address used for transmission. 0x000000000000FFFF is the broadcast address for the PAN. DL is not supported in API Mode. 0x0000000000000000 is the Coordinator’s 16-bit network address.	CRE	0 - 0xFFFFFFFF	0xFFFF(Coordinator) 0 (Router/End Device)
MY	**16-bit Network Address.** Get the 16-bit network address of the module.	CRE	0 - 0xFFFE [read-only]	0xFFFE
MP	**16-bit Parent Network Address.** Get the 16-bit parent network address of the module.	E	0 - 0xFFFE [read-only]	0xFFFE
NC	**Number of Children**. Read the number of end device children that have joined to the device. This command returns the number of child table entries currently in use.	CR	0 - 8	read-only
SH	**Serial Number High**. Read high 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled.	CRE	0 - 0xFFFFFFFF [read-only]	factory-set
SL	**Serial Number Low**. Read low 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled.	CRE	0 - 0xFFFFFFFF [read-only]	factory-set
NI	**Node Identifier.** Stores a string identifier. The register only accepts printable ASCII data. In AT Command Mode, a string can not start with a space. A carriage return ends the command. Command will automatically end when maximum bytes for the string have been entered. This string is returned as part of the ND (Node Discover) command. This identifier is also used with the DN (Destination Node) command.	CRE	20-Byte printable ASCII string	ASCII space character (0x20)
DD	**Device Type Identifier.** Stores a device type value. This value can be used to differentiate multiple XBee-based products.	CRE	0 - 0xFFFFFFFF [read-only]	0x20000
ZA[2]	**ZigBee Application Layer Addressing**. Set/read the Zigbee application layer addressing enabled attribute. If enabled, data packets will use the SE, DE, and CI commands to address Zigbee application layer source and destination endpoints, and the cluster ID fields in all data transmissions. ZA is only supported in the AT firmware.	CRE	0 - 1	0
SE[2]	**Source Endpoint**. Set/read the ZigBee application layer source endpoint value. If ZigBee application layer addressing is enabled (ZA command), this value will be used as the source endpoint for all data transmissions. SE is only supported in AT firmware.The default value 0xE8 (Data endpoint) is the Digi data endpoint	CRE	1 - 0xEF	0xE8

Table 7-02. Addressing Commands)

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
DE[2]	**Destination Endpoint**. Set/read Zigbee application layer destination ID value. If ZigBee application layer addressing is enabled (ZA command), this value will be used as the destination endpoint all data transmissions. DE is only supported in AT firmware.The default value (0xE8) is the Digi data endpoint.	CRE	0 - 0xEF	1 - 0xEF
CI[2]	**Cluster Identifier**. Set/read Zigbee application layer cluster ID value. If ZigBee application layer addressing is enabled (ZA command), this value will be used as the cluster ID for all data transmissions. CI is only supported in AT firmware.The default value0x11 (Transparent data cluster ID).	CRE	0 - 0xFF	0x11
BI[2]	**Binding Table Index**. Set/read the binding table index value. If this value is set to a valid binding table index, the addressing information at that index in the binding table will be used for all data transmissions. BI is only supported in AT firmware	CRE	0 - 0xFF	0xFF

1. Node types that support the command: C=Coordinator, R=Router, E=End Device
2. Command supported by modules using AT Command firmware only

Networking & Security

Table 7-03. Networking Commands

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
CH	**Operating Channel**. Read the channel number used for transmitting and receiving between RF modules. Uses 802.15.4 channel numbers. A value of 0 means the device has not joined a PAN and is not operating on any channel.	CRE	0, 0x0B-0x1A *(XBee)* 0, 0x0C – 0x18 (XBee-PRO)	[read-only]
ID	**PAN ID**. Set/Get the PAN (Personal Area Network) ID. ***Coordinator*** - Set the preferred Pan ID. Set ID = 0xFFFF to auto-select. ***Router / End Device*** - Set the desired Pan ID. When the device searches for a Coordinator, it attempts to only join to a parent that has a matching Pan ID. Set ID = 0xFFFF to join a parent operating on any Pan ID. Changes to ID should be written to non-volatile memory using the WR command.	CRE	0 - 0x3FFF, 0xFFFF	0x0234 (291d)
BH	**Broadcast Hops.** Set/Read the maximum number of hops for each broadcast data transmission. Setting this to 0 will use the maximum number of hops.	CRE	0 - 0x20	0
OP	**Operating PAN ID**. Read the PAN (Personal Area Network) ID. The OP value reflects the operating PAN ID that the module is running on. If ID < 0xFFFF, OP will equal ID.	CRE	0 - 0x3FFF	[read-only]
NT	**Node Discover Timeout.** Set/Read the amount of time a node will spend discovering other nodes when ND or DN is issued.	CRE	0x20 - 0xFF [x 100 msec]	0x3C (60d)
NO	**Network Discovery options.** Set/Read the options value for the network discovery command. The options bitfield value can change the behavior of the ND (network discovery) command and/or change what optional values are returned in any received ND responses or API node identification frames. Options include: 0x01 = Append DD value (to ND responses or API node identification frames) 002 = Local device sends ND response frame when ND is issued.	CRE	0 - 0x03 [bitfield]	0
ND	**Node Discover.** Discovers and reports all RF modules found. The following information is reported for each module discovered. MY<CR> SH<CR> SL<CR> NI<CR> (Variable length) PARENT_NETWORK ADDRESS (2 Bytes)<CR> DEVICE_TYPE<CR> (1 Byte: 0=Coord, 1=Router, 2=End Device) STATUS<CR> (1 Byte: Reserved) PROFILE_ID<CR> (2 Bytes) MANUFACTURER_ID<CR> (2 Bytes) <CR> After (NT * 100) milliseconds, the command ends by returning a <CR>. ND also accepts a Node Identifier (NI) as a parameter (optional). In this case, only a module that matches the supplied identifier will respond. If ND is sent through the API, each response is returned as a separate AT_CMD_Response packet. The data consists of the above listed bytes without the carriage return delimiters. The NI string will end in a "0x00" null character. The radius of the ND command is set by the BH command.	CRE	optional 20-Byte NI or MY value	--

Table 7-03. Networking Commands

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
DN	**Destination Node.** Resolves an NI (Node Identifier) string to a physical address (case-sensitive). The following events occur after the destination node is discovered: <AT Firmware> 1. DL & DH are set to the extended (64-bit) address of the module with the matching NI (Node Identifier) string. 2. OK (or ERROR)\r is returned. 3. Command Mode is exited to allow immediate communication <API Firmware> 1. The 16-bit network and 64-bit extended addresses are returned in an API Command Response frame. If there is no response from a module within (NT * 100) milliseconds or a parameter is not specified (left blank), the command is terminated and an "ERROR" message is returned. In the case of an ERROR, Command Mode is not exited. The radius of the DN command is set by the BH command.	CRE	up to 20-Byte printable ASCII string	--
SC	**Scan Channels**. Set/Read the list of channels to scan. ***Coordinator*** - Bit field list of channels to choose from prior to starting network. ***Router/End Device*** - Bit field list of channels that will be scanned to find a Coordinator/ Router to join. Changes to SC should be written using WR command. Bit (Channel): 0 (0x0B) 4 (0x0F) 8 (0x13) 12 (0x17) 1 (0x0C) 5 (0x10) 9 (0x14) 13 (0x18) 2 (0x0D) 6 (0x11) 10 (0x15) 14 (0x19) 3 (0x0E) 7 (0x12) 11 (0x16) 15 (0x1A) Note: Setting SC to include more than 12 continuous channels could cause data to be received on incorrect frequencies due to crosstalk issues with the EM250 at certain power levels. See Appendix E for details. Changing SC may result in not being able to communicate with long-range '-PRO' modules from Digi	CRE	**XBee** 1 - 0xFFFF [bitfield] **XBee-PRO** 2 - 0x3FFE [bitfield] (bits 0, 14, 15 not allowed)	0x1FFE
SD	**Scan Duration**. Set/Read the scan duration exponent. Changes to SD should be written using WR command. ***Coordinator*** - Duration of the Active and Energy Scans (on each channel) that are used to determine an acceptable channel and Pan ID for the Coordinator to startup on. ***Router / End Device*** - Duration of Active Scan (on each channel) used to locate an available Coordinator / Router to join during Association. Scan Time is measured as:(# Channels to Scan) * (2 ^ SD) * 15.36ms - The number of channels to scan is determined by the SC parameter. The XBee can scan up to 16 channels (SC = 0xFFFF). Sample Scan Duration times (13 channel scan): If SD = 0, time = 0.200 sec SD = 2, time = 0.799 sec SD = 4, time = 3.190 sec SD = 6, time = 12.780 sec	CRE	0 - 7 [exponent]	3
NJ	**Node Join Time**. Set/Read the time that a Coordinator/Router allows nodes to join. This value can be changed at run time without requiring a Coordinator or Router to restart. The time starts once the Coordinator or Router has started. The timer is reset on power-cycle or when NJ changes.	CR	0 – 0x40, 0xFF [x 1 sec]	0xFF (always allows joining)
JV	**Channel Verification**. Set/Read the channel verification parameter. If JV=1, and the network is an open network (NJ=0xFF), a router will verify the coordinator is on its operating channel when joining or coming up from a power cycle. If a coordinator is not detected, the router will leave its current channel and attempt to join a new PAN. If JV=0, the router will continue operating on its current channel even if a coordinator is not detected.	R	0 - Channel verification disabled 1 - Channel verification enabled	0
AR	**Aggregate Routing Notification**. Set/read time between consecutive aggregate route broadcast messages. If used, AR should be set on only one device to enable many-to-one routing to the device. Setting AR to 0 only sends one broadcast	CR	0 - 0xFF	0xFF
AI	**Association Indication**. Read information regarding last node join request: 0x00 - Successful completion - Coordinator started or Router/End Device found and joined with a parent. 0x21 - Scan found no PANs 0x22 - Scan found no valid PANs based on current SC and ID settings 0x23 - Valid Coordinator or Routers found, but they are not allowing joining (NJ expired) 0x27 - Node Joining attempt failed (typically due to incompatible security settings) 0x2A - Coordinator Start attempt failed' 0xFF - Scanning for a Parent 0x2B - Checking for an existing coordinator	CRE	0 - 0xFF [read-only]	--

Security

Table 7-04. Security Commands

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
EE	Encryption Enable. Set/Read the encryption enable setting.	CRE	0 - Encryption disabled 1 - Encryption enabled	0
EO	Encryption Options. Configure options for encryption. Unused option bits should be set to 0. Options include: 0x01 - Send the security key unsecured over-the-air during joins 0x02 - Use trust center	CRE	0 - 0xFF	
KY	Encryption Key. Set the 128-bit AES encryption key. This command is read-only; KY cannot be read.	CRE	0 - 0xFFFFFFFFFFFFFFFF	0

RF Interfacing

Table 7-05. RF Interfacing Commands

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
PL	**Power Level**. Select/Read the power level at which the RF module transmits conducted power.	CRE	**XBee** (boost mode disabled) 0 = -8 dBm 1 = -4 dBm 2 = -2 dBm 3 = 0 dBm 4 = +2 dBm **XBee-PRO** 4 = 18 dBm **XBee-PRO (International Variant)** 4 = 10dBm	4
PM	**Power Mode**. Set/read the power mode of the device. Enabling boost mode will improve the receive sensitivity by 1dB and increase the transmit power by 2dB Note: Enabling boost mode on the XBee-PRO will not affect the output power. Boost mode imposes a slight increase in current draw. See section 1.2 for details.	CRE	0-1, 0= -Boost mode disabled, 1= Boost mode enabled.	1
DB	**Received Signal Strength**. This command reports the received signal strength of the last received RF data packet. The DB command only indicates the signal strength of the last hop. It does not provide an accurate quality measurement for a multihop link. DB can be set to 0 to clear it.			

1. Node types that support the command: C = Coordinator, R = Router, E = End Device

Serial Interfacing (I/O)

Table 7-06. Serial Interfacing Commands

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
AP[2]	**API Enable**. Enable API Mode. The AP parameter is only applicable when using modules that contain the following firmware versions:1.1xx (coordinator), 1.3xx (router/end device)	CRE	1 - 2 1 = API-enabled 2 = API-enabled (w/escaped control characters)	1
AO[2]	**API Options**. Configure options for API. Current options select the type of receive API frame to send out the Uart for received RF data packets.	CRE	0 - Default receive API indicators enabled 1 - Explicit Rx data indicator API frame enabled (0x91)	0
BD	**Interface Data Rate**. Set/Read the serial interface data rate for communication between the module serial port and host. Any value above 0x07 will be interpreted as an actual baud rate. When a value above 0x07 is sent, the closest interface data rate represented by the number is stored in the BD register.	CRE	0 - 7 (standard baud rates) 0 = 1200 bps 1 = 2400 2 = 4800 3 = 9600 4 = 19200 5 = 38400 6 = 57600 7 = 115200 0x80 - 0x38400 (non-standard rates)	3

© 2008 Digi International, Inc.

Table 7-06. Serial Interfacing Commands

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
NB	**Serial Parity**. Set/Read the serial parity setting on the module.	CRE	0 = No parity 1 = Even parity 2 = Odd parity 3 = Mark parity	0
RO	**Packetization Timeout**. Set/Read number of character times of inter-character silence required before packetization. Set (RO=0) to transmit characters as they arrive instead of buffering them into one RF packet.	CRE	0 - 0xFF [x character times]	3
D7	**DIO7 Configuration**. Select/Read options for the DIO7 line of the RF module.	CRE	0 = Disabled 1 = CTS Flow Control 3 = Digital input 4 = Digital output, low 5 = Digital output, high 6 = RS-485 transmit enable (low enable) 7 = RS-485 transmit enable (high enable)	1
D6	**DIO6 Configuration.** Configure options for the DIO6 line of the RF module.	CRE	0 - Disabled 1 - RTS Flow Control	0

1. Node types that support the command: C = Coordinator, R = Router, E = End Device
2. Command supported by modules using API firmware only

I/O Commands

Table 7-07. I/O Commands

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
IS	**Force Sample** Forces a read of all enabled digital and analog input lines.	CRE	--	--
1S	**XBee Sensor Sample**. Forces a sample to be taken on an XBee Sensor device. This command can only be issued to an XBee sensor device using an API remote command.	RE	-	-
IR	**IO Sample Rate**. Set/Read the IO sample rate to enable periodic sampling. For periodic sampling to be enabled, IR must be set to a non-zero value, and at least one module pin must have analog or digital IO functionality enabled (see D0-D8, P0-P2 commands). The sample rate is measured in milliseconds.	CRE	0 - 0xFFFF (ms)	0
IC	I**O Digital Change Detection**. Set/Read the digital IO pins to monitor for changes in the IO state. IC works with the individual pin configuration commands (D0-D8, P0-P2). If a pin is enabled as a digital input/output, the IC command can be used to force an immediate IO sample transmission when the DIO state changes. IC is a bitmask that can be used to enable or disable edge detection on individual channels. Unused bits should be set to 0. Bit (IO pin): 0 (DIO0)4 (DIO4)8 (DIO8) 1 (DIO1) 5 (DIO5) 9 (DIO9) 2 (DIO2) 6 (DIO6) 10 (DIO10) 3 (DIO3) 7 (DIO7) 11 (DIO11)	CRE	: 0 - 0xFFFF	0
P0	**PWM0 Configuration**. Select/Read function for PWM0.	CRE	0 = Disabled 1 = RSSI PWM 3 - Digital input, monitored 4 - Digital output, default low 5 - Digital output, default high	1
P1	**DIO11 Configuration**. Configure options for the DIO11 line of the RF module.	CRE	0 - Unmonitored digital input 3- Digital input, monitored 4- Digital output, default low 5- Digital output, default high	0
P2	**DIO12 Configuration**. Configure options for the DIO12 line of the RF module.	CRE	0 - Unmonitored digital input 3- Digital input, monitored 4- Digital output, default low 5- Digital output, default high	0

© 2008 Digi International, Inc.

Table 7-07. I/O Commands

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
P3	**DIO13 Configuration**. Set/Read function for DIO13. This command is not yet supported.	CRE	0, 3-5 0 – Disabled 3 – Digital input 4 – Digital output, low 5 – Digital output, high	
D0	**AD0/DIO0 Configuration**. Select/Read function for AD0/DIO0.	CRE	0-5 0 – Disabled 1 - Node identification button enabled 2 - Analog input, single ended 3 – Digital input 4 – Digital output, low 5 – Digital output, high	1
D1	**AD1/DIO1 Configuration**. Select/Read function for AD1/DIO1.	CRE	0, 2-5 0 – Disabled 2 - Analog input, single ended 3 – Digital input 4 – Digital output, low 5 – Digital output, high	0
D2	AD2/DIO2 Configuration. Select/Read function for AD2/DIO2.	CRE	0, 2-5 0 – Disabled 2 - Analog input, single ended 3 – Digital input 4 – Digital output, low 5 – Digital output, high	0
D3	**AD3/DIO3 Configuration**. Select/Read function for AD3/DIO3.	CRE	0, 2-5 0 – Disabled 2 - Analog input, single ended 3 – Digital input 4 – Digital output, low 5 – Digital output, high	0
D4	**DIO4 Configuration**. Select/Read function for DIO4.	CRE	0, 3-5 0 – Disabled 3 – Digital input 4 – Digital output, low 5 – Digital output, high	0
D5	**DIO5 Configuration**. Configure options for the DIO5 line of the RF module.	CRE	0 = Disabled 1 = Associated indication LED 3 = Digital input 4 = Digital output, default low 5 = Digital output, default high	1
LT	**Assoc LED Blink Time**. Set/Read the Associate LED blink time. If the Associate LED functionality is enabled (D5 command), this value determines the on and off blink times for the LED when the module has joined a network. If LT=0, the default blink rate will be used (500ms coordinator, 250ms router/end device). For all other LT values, LT is measured in 10ms.	CRE	0x14 - 0xFF (200 - 2550 ms)	0
D8	**DIO8 Configuration**. Set/Read function for DIO8. This command is not yet supported.	CRE	0, 3-5 0 – Disabled 3 – Digital input 4 – Digital output, low 5 – Digital output, high	

© 2008 Digi International, Inc.

Table 7-07. I/O Commands

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
PR	Set/read the bit field that configures the internal pull-up resistor status for the I/O lines. "1" specifies the pull-up resistor is enabled. "0" specifies no pullup.(30k pull-up resistors) Bits:" 0 - DIO4 (Pin 11) 1 - AD3 / DIO3 (Pin 17) 2 - AD2 / DIO2 (Pin 18) 3 - AD1 / DIO1 (Pin 19) 4 - AD0 / DIO0 (Pin 20) 5 - RTS / DIO6 (Pin 16) 6 - DTR / Sleep Request / DIO8 (Pin 9) 7 - DIN / Config (Pin 3) 8 - Associate / DIO5 (Pin 15) 9 - On/Sleep / DIO9 (Pin 13) 10 - DIO12 (Pin 4) 11 - PWM0 / RSSI / DIO10 (Pin 6) 12 - PWM1 / DIO11 (Pin 7)	CRE	0 - 0x1FFF	0 - 0x1FFF
RP	**RSSI PWM Timer.** Time RSSI signal will be output after last transmission. When RP = 0xFF, output will always be on.	CRE	0 - 0xFF [x 100 ms]	0x28 (40d)
CB	**Commissioning Pushbutton**. This command can be used to simulate commissioning button presses in software. The parameter value should be set to the number of button presses to be simulated. For example, sending the ATCB1 command will execute the action associated with 1 commissioning button press. (See D0 command).	CRE		

Diagnostics

Table 7-08. Diagnostics Commands

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
VR	**Firmware Version**. Read firmware version of the module.	CRE	0 - 0xFFFF [read-only]	Factory-set
HV	**Hardware Version**. Read hardware version of the module.	CRE	0 - 0xFFFF [read-only]	Factory-set
%V	**Supply Voltage.** Reads the voltage on the Vcc pin. To convert the reading to a mV reading, divide the read value by 1023 and multiply by 1200. A %V reading of 0x8FE (2302 decimal) represents 2700mV or 2.70V.	CRE	-	-

1. Node types that support the command:C = Coordinator, R = Router, E = End Device

AT Command Options

Table 7-09. AT Command Options Commands

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
CT[2]	**Command Mode Timeout.** Set/Read the period of inactivity (no valid commands received) after which the RF module automatically exits AT Command Mode and returns to Idle Mode.	CRE	2 - 0x028F [x 100 ms]	0x64 (100d)
CN[2]	**Exit Command Mode.** Explicitly exit the module from AT Command Mode.	CRE	--	--
GT[2]	**Guard Times**. Set required period of silence before and after the Command Sequence Characters of the AT Command Mode Sequence (GT + CC + GT). The period of silence is used to prevent inadvertent entrance into AT Command Mode.	CRE	1 - 0x0CE4 [x 1 ms] (max of 3.3 decimal sec)	0x3E8 (1000d)
CC[2]	**Command Sequence Character**. Set/Read the ASCII character value to be used between Guard Times of the AT Command Mode Sequence (GT + CC + GT). The AT Command Mode Sequence enters the RF module into AT Command Mode. CC command is only applicable when using modules that contain the following "AT Command" firmware versions: 8.0xx (Coordinator), 8.2xx (Router), 8.4xx (End Device)	CRE	0 - 0xFF	0x2B ('+' ASCII)

1. Node types that support the command: C = Coordinator, R = Router, E = End Device
2. Command supported by modules using AT Command firmware only

© 2008 Digi International, Inc.

Sleep Commands

Table 7-010. Sleep Commands

AT Command	Name and Description	Node Type[1]	Parameter Range	Default
SM	**Sleep Mode** Sets the sleep mode on the RF module	RE	0-Sleep disabled 1-Pin sleep enabled 4-Cyclic sleep enabled Note: When SM=0, the device operates as a router. When SM changes to a non-zero value, the router leaves the network and rejoins as an end device. **Only end devices can sleep**	0
SN	**Number of Sleep Periods.** Sets the number of sleep periods to not assert the On/Sleep pin on wakeup if no RF data is waiting for the end device. This command allows a host application to sleep for an extended time if no RF data is present	RE	1 - 0xFFFF	1
SP	**Sleep Period.** This value determines how long the end device will sleep at a time, up to 28 seconds. (The sleep time can effectively be extended past 28 seconds using the SN command.) On the parent, this value determines how long the parent will buffer a message for the sleeping end device. It should be set at least equal to the longest SP time of any child end device.	CRE	0x20 - 0xAF0 x 10ms (Quarter second resolution)	0x20
ST	**Time Before Sleep** Sets the time before sleep timer on an end device.The timer is reset each time serial or RF data is received. Once the timer expires, an end device may enter low power operation. Applicable for cyclic sleep end devices only.	RE	1 - 0xFFFE (x 1ms)	0x1388 (5 seconds)
SO Command	**Sleep Options**. Configure options for sleep. Unused option bits should be set to 0. Sleep options include: 0x02 - Always wake for ST time 0x04 - Sleep entire SN * SP time Sleep options should not be used for most applications. See Sleep Mode chapter for more information.	E	0 - 0xFF	0

8. Manufacturing Support

8.1. Interoperability with other EM250 Devices

The XBee module can interoperate with other EM250-based devices. The following should be considered when communicating between a Digi XBee module and another EM250-based device.

The XBee firmware inserts 8 bytes at the beginning of the data payload that represent the 64-bit address of the source module. Custom devices that transmit to an XBee or receive data from an XBee should make provisions to manage these 8 address bytes in the payload. Data packets destined for an XBee module should include the source address of the sending device with the most significant byte copied first.

8.1.1. Crosstalk Issue

The EM250 suffers from a crosstalk issue where data received on a channel that is 12 channels above or below the current channel appears to be received on the current channel. RF data is detected on the erroneous channel only if the signal level of the received RF data is around -30dBm (XBee devices separated by a few feet). For example, RF traffic on channel 0x0B could be seen on channels 0x0B and 0x17 if its detected signal level is around -30dBm. This could result in routers or end devices joining a PAN, but reporting an incorrect channel.

The problem can best be avoided by setting the SC (scan channels) bitmask to only include 12 continuous channels. If SC is left at its default value (0x1FFE), the crosstalk issue will never occur.

8.2. Customizing XBee Default Parameters

Once module parameters are determined, Digi can manufacture modules with specific customer-defined configurations. These custom configurations can lock in a firmware version or set command values when the modules are manufactured, eliminating the need for customers to adjust module parameters on arrival. Contact Digi to create a custom configuration.

8.3. XBee EM250 Pin Mappings

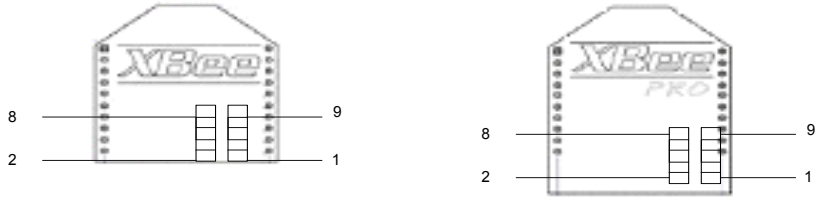
The following table shows how the GPIO pins on the EM250 map to pins on the XBee ZNet 2.5 module:

Table 8-011.

XBee ZNet 2.5 Module Pin Number	EM250 Pin Number	EM250 GPIO
1	-	
2	32	9
3	33	10
4	31	8
5	13	
6	41	15
7	40	16
8	-	-
9	42	14
10	-	-
11	43	13
12	20	12
13	25	3
14	-	-
15	21	0

Table 8-011.

XBee ZNet 2.5 Module Pin Number	EM250 Pin Number	EM250 GPIO
16	19	11
17	30	7
18	29	6
19	27	5
20	26	4



This figure shows the orientation of the insight port header.

Pin Number	Pin Name
1	VBRD
2	SIF-MISO
3	Ground
4	SIF-MOSI
5	Ground
6	SIF-CLOCK
7	SIF-LOAD
8	RESET
9	PTI-EN
10	PTI-DATA

8.4. XBee ZNet 2.5 Custom Bootloader

XBee ZNet 2.5 modules use a modified version of Ember's boot loader. This bootloader version supports a custom entry mechanism that uses module pins DIN (pin 3), $\overline{\text{DTR}}$ / SLEEP_RQ (pin 9), and $\overline{\text{RTS}}$ (pin 16). To invoke the boot loader, do the following:

1. Set $\overline{\text{DTR}}$ / SLEEP_RQ low (TTL 0V) and RTS high.

2. Send a serial break to the DIN pin and power cycle or reset the module.

3. When the module powers up, $\overline{\text{DTR}}$ / SLEEP_RQ and DIN should be low (TTL 0V) and RTS should be high.

4. Terminate the serial break and send a carriage return at 115200bps to the module.

5. If successful, the module will send the Ember boot loader menu out the DOUT pin at 115200bps.

6. Commands can be sent to the boot loader at 115200bps.

Note: Hardware flow control should be disabled when entering and communicating with the EM250 bootloader.

© 2008 Digi International, Inc.

8.5. Programming XBee ZNet 2.5 Modules

Firmware on the XBee ZNet 2.5 modules can be upgraded using the Digi X-CTU program to interface with the DIN and DOUT serial lines, or with an InSight programmer device via InSight header.

An external application can upload firmware to the XBee ZNet 2.5 modules by doing the following:

1. Enter bootloader mode as described in section 9.4

2. The application should look for the bootloader "BL >" prompt to be sent out the UART to ensure the bootloader is active.

3. Send an ascii "1" to initiate a firmware update.

4. After sending a "1", the EM250 waits for an XModem CRC upload of an .ebl image over the serial line. The .ebl file must be sent to the EM250 in order.

If no transaction is initiated within 60 seconds, the bootloader times out and returns to the menu. If the upload is interrupted with a power cycle or reset event, the EM250 will detect an invalid application image and enter bootloader mode. The entire ebl image should be uploaded again to recover. If an error occurs while uploading, the EM250 bootloader returns an error code from the following table:

Table 8-012.

Hex Error Code	Description
0x21	The bootloader encountered an error while trying to parse the Start of Header (SOH) character in the XModem frame.
0x22	The bootloader detected an invalid checksum in the XModem frame.
0x23	The bootloader encountered an error while trying to parse the high byte of the CRC in the XModem frame.
0x24	The bootloader encountered an error while trying to parse the low byte of the CRC in the XModem frame.
0x25	The bootloader encountered an error in the sequence number of the current XModem frame.
0x26	The frame that the bootloader was trying to parse was deemed incomplete (some bytes missing or lost).
0x27	The bootloader encountered a duplicate of the previous XModem frame.
0x41	No .ebl header was received when expected.
0x42	Header failed CRC.
0x43	File failed CRC.
0x44	Unknown tag detected in .ebl image.
0x45	Invalid .ebl header signature.
0x46	Trying to flash odd number of bytes.
0x47	Indexed past end of block buffer.
0x48	Attempt to overwrite bootloader flash.
0x49	Attempt to overwrite SIMEE flash.
0x4A	Flash erase failed.
0x4B	Flash write failed.
0x4C	End tag CRC wrong length.
0x4D	Received data before query request/response

© 2008 Digi International, Inc.

8.6. Developing Custom Firmware

Designers can implement custom firmware projects on the XBee module and upload the firmware using X-CTU or one of Ember's programming tools. The X-CTU can upload firmware onto an XBee as long as the original bootloader is not erased (see section 8.4).

For some applications, it may be necessary to determine if the board is an XBee or an XBee-PRO flavor. The GPIO1 pin is used to identify the module type – it is grounded on the XBee, and unconnected on the XBee-PRO. To determine if a module is an XBee or an XBee-PRO, do the following:

- GPIO_DIRCLRL = GPIO(1); // Set GPIO1 as an input
- GPIO_PUL |= GPIO(1); // Enable GPIO1 pullup resistor
- ModuleIsXBeePro = (GPIO_INL & GPIO(1)); // Read GPIO1. If high, XBee-PRO. If low, XBee.

Custom applications should call emberSetTxPowerMode() with the appropriate parameters to configure the RF path correctly.

- Applications running on the XBee should callemberSetTxPower-Mode(EMBER_TX_POWER_MODE_DEFAULT) or emberSetTxPower-Mode(EMBER_TX_POWER_MODE_BOOST).
- Applications running on the XBee-PRO should call emberSetTxPower-Mode(EMBER_TX_POWER_MODE_ALTERNATE) or emberSetTxPower-Mode(EMBER_TX_POWER_MODE_BOOST_AND_ALTERNATE).

8.7. Design Considerations for Digi Drop-in Networking

XBee/XBee-PRO ZNet 2.5 embedded RF modules contain a variety of features that allow for interoperability with Digi's full line of Drop-in Networking products. Interoperability with other "DIN" products can offer these advantages:

- Add IP-connectivity to your network via Cellular, Ethernet or WiFi with a ConnectPort X Gateway.
- Extend the range of your network with the XBee Wall Router.
- Make deployment easy by enabling the Commissioning Pushbutton (pin 20) and Associate LED (pin 15) to operate with the Network Commissioning Tool software.
- Interface with standard RS-232, USB, Analog & Digital I/O, RS-485, and other industrial devices using XBee Adapters.
- Monitor and manage your network securely from remote locations with Connectware Manager software.

We encourage you to contact our technical representatives for consideration, implementation, or design review of your product for interoperability with Digi's Drop-in Networking solutions.

© 2008 Digi International, Inc.

Appendix A: Definitions

Definitions

Table A-01. Terms and Definitions

ZigBee Node Types	
Coordinator	A node that has the unique function of forming a network. The coordinator is responsible for establishing the operating channel and PAN ID for an entire network. Once established, the coordinator can form a network by allowing routers and end devices to join to it. Once the network is formed, the coordinator functions like a router (it can participate in routing packets and be a source or destination for data packets). -- One coordinator per PAN -- Establishes/Organizes PAN -- Can route data packets to/from other nodes -- Can be a data packet source and destination -- Mains-powered Refer to the XBee ZNet 2.5 coordinator section for more information.
Router	A node that creates/maintains network information and uses this information to determine the best route for a data packet. A router must join a network before it can allow other routers and end devices to join to it. A router can participate in routing packets and is intended to be a mains-powered node. -- Several routers can operate in one PAN -- Can route data packets to/from other nodes -- Can be a data packet source and destination -- Mains-powered Refer to the XBee ZNet 2.5 router section for more information.
End device	End devices must always interact with their parent to receive or transmit data. (See 'joining definition.) They are intended to sleep periodically and therefore have no routing capacity. An end device can be a source or destination for data packets but cannot route packets. End devices can be battery-powered and offer low-power operation. -- Several end devices can operate in one PAN -- Can be a data packet source and destination -- All messages are relayed through a coordinator or router -- Lower power modes
ZigBee Protocol	
PAN	Personal Area Network - A data communication network that includes a coordinator and one or more routers/end devices.

Table A-01. Terms and Definitions

Joining	The process of a node becoming part of a ZigBee PAN. A node becomes part of a network by joining to a coordinator or a router (that has previously joined to the network). During the process of joining, the node that allowed joining (the parent) assigns a 16-bit address to the joining node (the child).
Network Address	The 16-bit address assigned to a node after it has joined to another node. The coordinator always has a network address of 0.
Operating Channel	The frequency selected for data communications between nodes. The operating channel is selected by the coordinator on power-up.
Energy Scan	A scan of RF channels that detects the amount of energy present on the selected channels. The coordinator uses the energy scan to determine the operating channel.
Route Request	Broadcast transmission sent by a coordinator or router throughout the network in attempt to establish a route to a destination node.
Route Reply	Unicast transmission sent back to the originator of the route request. It is initiated by a node when it receives a route request packet and its address matches the Destination Address in the route request packet.
Route Discovery	The process of establishing a route to a destination node when one does not exist in the Routing Table. It is based on the AODV (Ad-hoc On-demand Distance Vector routing) protocol.
ZigBee Stack	ZigBee is a published specification set of high-level communication protocols for use with small, low-power modules. The ZigBee stack provides a layer of network functionality on top of the 802.15.4 specification. For example, the mesh and routing capabilities available to ZigBee solutions are absent in the 802.15.4 protocol.

Appendix B: Agency Certifications

United States FCC

The XBee ZNet 2.5 RF Module complies with Part 15 of the FCC rules and regulations. Compliance with the labeling requirements, FCC notices and antenna usage guidelines is required.

To fufill FCC Certification, the OEM must comply with the following regulations:

1.The system integrator must ensure that the text on the external label provided with this device is placed on the outside of the final product. [Figure A-01]

2.XBee ZNet 2.5 RF Module may only be used with antennas that have been tested and approved for use with this module [refer to the antenna tables in this section].

OEM Labeling Requirements

WARNING: The Original Equipment Manufacturer (OEM) must ensure that FCC labeling requirements are met. This includes a clearly visible label on the outside of the final product enclosure that displays the contents shown in the figure below.

Required FCC Label for OEM products containing the XBee ZNet 2.5 RF Module

> Contains FCC ID: OUR-XBEE2*
>
> The enclosed device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (*i.*) this device may not cause harmful interference and (*ii.*) this device must accept any interference received, including interference that may cause undesired operation.

Required FCC Label for OEM products containing the XBee PRO ZNet 2.5 RF Module

> Contains FCC ID:MCQ-XBEEPRO2*
>
> The enclosed device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (*i.*) this device may not cause harmful interference and (*ii.*) this device must accept any interference received, including interference that may cause undesired operation.

FCC Notices

IMPORTANT: The XBee ZNet 2.5 and XBee PRO ZNet 2.5 RF Module have been certified by the FCC for use with other products without any further certification (as per FCC section 2.1091). Modifications not expressly approved by Digi could void the user's authority to operate the equipment.

IMPORTANT: OEMs must test final product to comply with unintentional radiators (FCC section 15.107 & 15.109) before declaring compliance of their final product to Part 15 of the FCC Rules.

IMPORTANT: The RF module has been certified for remote and base radio applications. If the module will be used for portable applications, the device must undergo SAR testing.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures: Re-orient or relocate the receiving antenna, Increase the separation between the equipment and receiver, Connect equipment and receiver to outlets on different circuits, or Consult the dealer or an experienced radio/TV technician for help.

FCC-Approved Antennas (2.4 GHz)

The XBee ZNet 2.5 and XBee PRO ZNet 2.5 RF Module can be installed utilizing antennas and cables constructed with standard connectors (Type-N, SMA, TNC, etc.) if the installation is performed professionally and according to FCC guidelines. For installations not performed by a professional, non-standard connectors (RPSMA, RPTNC, etc.) must be used.

The modules are FCC approved for fixed base station and mobile applications on channels 0x0B-0x1A for Xbee Series2 and on channels 0x0B - 0x18 for Xbee ZNet-PRO 2.5 . If the antenna is mounted at least 20cm (8 in.) from nearby persons, the application is considered a mobile application. Antennas not listed in the table must be tested to comply with FCC Section 15.203 (Unique Antenna Connectors) and Section 15.247 (Emissions).

XBee ZNet 2.5 RF Modules: XBee ZNet 2.5 RF Modules have been tested and approved for use with all the antennas listed in the tables below. (Cable-loss IS required when using gain antennas as shown below.)

Table A-01. antennas approved for use with the XBee ZNet 2.5 RF Modules

YAGI CLASS ANTENNAS					
Part Number	**Type (Description)**	**Gain**	**Application***	**Min. Separation Required**	**Cable-loss**
A24-Y6NF	Yagi (6-element)	8.8 dBi	Fixed	2 m	N/A
A24-Y7NF	Yagi (7-element)	9.0 dBi	Fixed	2 m	N/A
A24-Y9NF	Yagi (9-element)	10.0 dBi	Fixed	2 m	N/A
A24-Y10NF	Yagi (10-element)	11.0 dBi	Fixed	2 m	N/A
A24-Y12NF	Yagi (12-element)	12.0 dBi	Fixed	2 m	N/A
A24-Y13NF	Yagi (13-element)	12.0 dBi	Fixed	2 m	N/A
A24-Y15NF	Yagi (15-element)	12.5 dBi	Fixed	2 m	N/A
A24-Y16NF	Yagi (16-element)	13.5 dBi	Fixed	2 m	N/A
A24-Y16RM	Yagi (16-element, RPSMA connector)	13.5 dBi	Fixed	2 m	N/A
A24-Y18NF	Yagi (18-element)	15.0 dBi	Fixed	2 m	N/A
OMNI-DIRECTIONAL ANTENNAS					
Part Number	**Type (Description)**	**Gain**	**Application***	**Min. Separation Required**	**Cable-loss**
A24-C1	Surface Mount integral chip	-1.5 dBi	Fixed/Mobile	20 cm	N/A
A24-F2NF	Omni-directional (Fiberglass base station)	2.1 dBi	Fixed/Mobile	20 cm	N/A
A24-F3NF	Omni-directional (Fiberglass base station)	3.0 dBi	Fixed/Mobile	20 cm	N/A
A24-F5NF	Omni-directional (Fiberglass base station)	5.0 dBi	Fixed/Mobile	20 cm	N/A
A24-F8NF	Omni-directional (Fiberglass base station)	8.0 dBi	Fixed	2 m	N/A
A24-F9NF	Omni-directional (Fiberglass base station)	9.5 dBi	Fixed	2 m	N/A
A24-F10NF	Omni-directional (Fiberglass base station)	10.0 dBi	Fixed	2 m	N/A
A24-F12NF	Omni-directional (Fiberglass base station)	12.0 dBi	Fixed	2 m	N/A
A24-F15NF	Omni-directional (Fiberglass base station)	15.0 dBi	Fixed	2 m	N/A
A24-W7NF	Omni-directional (Base station)	7.2 dBi	Fixed	2 m	N/A
A24-M7NF	Omni-directional (Mag-mount base station)	7.2 dBi	Fixed	2 m	N/A
PANEL CLASS ANTENNAS					
Part Number	**Type (Description)**	**Gain**	**Application***	**Min. Separation Required**	**Cable-loss**
A24-P8SF	Flat Panel	8.5 dBi	Fixed	2 m	N/A
A24-P8NF	Flat Panel	8.5 dBi	Fixed	2 m	N/A
A24-P13NF	Flat Panel	13.0 dBi	Fixed	2 m	N/A
A24-P14NF	Flat Panel	14.0 dBi	Fixed	2 m	N/A
A24-P15NF	Flat Panel	15.0 dBi	Fixed	2 m	N/A
A24-P16NF	Flat Panel	16.0 dBi	Fixed	2 m	N/A
A24-P19NF	Flat Panel	19.0 dBi	Fixed	2m	1.5 dB

Table A-02. antennas approved for use with the XBee-PRO ZNet 2.5 RF Modules

YAGI CLASS ANTENNAS					
Part Number	**Type (Description)**	**Gain**	**Application***	**Min. Separation Required**	**Cable-loss**
A24-Y6NF	Yagi (6-element)	8.8 dBi	Fixed	2 m	7.8dB
A24-Y7NF	Yagi (7-element)	9.0 dBi	Fixed	2 m	8 dB
A24-Y9NF	Yagi (9-element)	10.0 dBi	Fixed	2 m	9 dB
A24-Y10NF	Yagi (10-element)	11.0 dBi	Fixed	2 m	10 dB
A24-Y12NF	Yagi (12-element)	12.0 dBi	Fixed	2 m	11 dB
A24-Y13NF	Yagi (13-element)	12.0 dBi	Fixed	2 m	11 dB
A24-Y15NF	Yagi (15-element)	12.5 dBi	Fixed	2 m	11.5 dB
A24-Y16NF	Yagi (16-element)	13.5 dBi	Fixed	2 m	12.5 dB
A24-Y16RM	Yagi (16-element, RPSMA connector)	13.5 dBi	Fixed	2 m	12.5 dB
A24-Y18NF	Yagi (18-element)	15.0 dBi	Fixed	2 m	14 dB
OMNI-DIRECTIONAL ANTENNAS					
Part Number	**Type (Description)**	**Gain**	**Application***	**Min. Separation Required**	**Cable-loss**
A24-C1	Surface Mount integral chip	-1.5dBi	Fixed/Mobile	20 cm	-
A24-F2NF	Omni-directional (Fiberglass base station)	2.1 dBi	Fixed/Mobile	20 cm	-
A24-F3NF	Omni-directional (Fiberglass base station)	3.0 dBi	Fixed/Mobile	20 cm	.3 dB
A24-F5NF	Omni-directional (Fiberglass base station)	5.0 dBi	Fixed/Mobile	20 cm	2.3 dB
A24-F8NF	Omni-directional (Fiberglass base station)	8.0 dBi	Fixed	2 m	5.3 dB
A24-F9NF	Omni-directional (Fiberglass base station)	9.5 dBi	Fixed	2 m	6.8 dB
A24-F10NF	Omni-directional (Fiberglass base station)	10.0 dBi	Fixed	2 m	7.3 dB
A24-F12NF	Omni-directional (Fiberglass base station)	12.0 dBi	Fixed	2 m	9.3dB
A24-F15NF	Omni-directional (Fiberglass base station)	15.0 dBi	Fixed	2 m	12.3dB
A24-W7NF	Omni-directional (Base station)	7.2 dBi	Fixed	2 m	4.5 dB
A24-M7NF	Omni-directional (Mag-mount base station)	7.2 dBi	Fixed	2 m	4.5 dB
PANEL CLASS ANTENNAS					
Part Number	**Type (Description)**	**Gain**	**Application***	**Min. Separation Required**	**Cable-loss**
A24-P8SF	Flat Panel	8.5 dBi	Fixed	2 m	8.2 dB
A24-P8NF	Flat Panel	8.5 dBi	Fixed	2 m	82 dB
A24-P13NF	Flat Panel	13.0 dBi	Fixed	2 m	12.7 dB
A24-P14NF	Flat Panel	14.0 dBi	Fixed	2 m	13.7 dB
A24-P15NF	Flat Panel	15.0 dBi	Fixed	2 m	14.7 dB
A24-P16NF	Flat Panel	16.0 dBi	Fixed	2 m	15.7 dB
A24-P19NF	Flat Panel	19.0 dBi	Fixed	2m	18.7 dB

*** If using the RF module in a portable application** (For example - If the module is used in a handheld device and the antenna is less than 20cm from the human body when the device is in operation): The integrator is responsible for passing additional SAR (Specific Absorption Rate) testing based on FCC rules 2.1091 and FCC Guidelines for Human Exposure to Radio Frequency Electromagnetic Fields, OET Bulletin and Supplement C. The testing results will be submitted to the FCC for approval prior to selling the integrated unit. The required SAR testing measures emissions from the module and how they affect the person.

RF Exposure

WARNING: To satisfy FCC RF exposure requirements for mobile transmitting devices, a separation distance of 20 cm or more should be maintained between the antenna of this device and persons during device operation. To ensure compliance, operations at closer than this distance are not recommended. The antenna used for this transmitter must not be co-located in conjunction with any other antenna or transmitter.

The preceding statement must be included as a CAUTION statement in OEM product manuals in order to alert users of FCC RF Exposure compliance.

Europe (ETSI)

The XBee ZNet 2.5 RF Module has been certified for use in several European countries. For a complete list, refer to www.digi.com

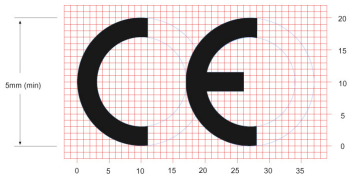
If the XBee ZNet 2.5 RF Modules are incorporated into a product, the manufacturer must ensure compliance of the final product to the European harmonized EMC and low-voltage/safety standards. A Declaration of Conformity must be issued for each of these standards and kept on file as described in Annex II of the R&TTE Directive.

Furthermore, the manufacturer must maintain a copy of the XBee ZNet 2.5 user manual documentation and ensure the final product does not exceed the specified power ratings, antenna specifications, and/or installation requirements as specified in the user manual. If any of these specifications are exceeded in the final product, a submission must be made to a notified body for compliance testing to all required standards.

OEM Labeling Requirements

The 'CE' marking must be affixed to a visible location on the OEM product.

Figure B-01. CE Labeling Requirements



The CE mark shall consist of the initials "CE" taking the following form:

- If the CE marking is reduced or enlarged, the proportions given in the above graduated drawing must be respected.
- The CE marking must have a height of at least 5mm except where this is not possible on account of the nature of the apparatus.
- The CE marking must be affixed visibly, legibly, and indelibly.

Restrictions

Power Output: The power output of the XBee ZNet 2.5 RF Module must not exceed 10 dBm. The power level is set using the PL command and the PL parameter must equal "0" (10 dBm).

France: France imposes restrictions on the 2.4 GHz band. Go to www.art-telecom.Fr or contact MaxStream for more information.

Norway: Norway prohibits operation near Ny-Alesund in Svalbard. More information can be found at the Norway Posts and Telecommunications site (www.npt.no).

Declarations of Conformity

Digi has issued Declarations of Conformity for the XBee ZNet 2.5 RF Modules concerning emissions, EMC and safety. Files are located in the 'documentation' folder of the Digi CD.

Important Note

Digi does not list the entire set of standards that must be met for each country. Digi customers assume full responsibility for learning and meeting the required guidelines for each country in their distribution market. For more information relating to European compliance of an OEM product incorporating the XBee ZNet 2.5 RF Module, contact Digi, or refer to the following web sites:

CEPT ERC 70-03E - Technical Requirements, European restrictions and general requirements: Available at www.ero.dk/.

R&TTE Directive - Equipment requirements, placement on market: Available at www.ero.dk/.

Approved Antennas

When integrating high-gain antennas, European regulations stipulate EIRP power maximums. Use the following guidelines to determine which antennas to design into an application.

XBee ZNet 2.5 OEM Module

The following antennas types have been tested and approved for use with the XBee ZNet 2.5 Module:

Antenna Type: Yagi

RF module was tested and approved with 15 dBi antenna gain with 1 dB cable-loss (EIRP Maimum of 14 dBm). Any Yagi type antenna with 14 dBi gain or less can be used with no cable-loss.

Antenna Type: Omni-Directional

RF module was tested and approved with 15 dBi antenna gain with 1 dB cable-loss (EIRP Maimum of 14 dBm). Any Omni-Directional type antenna with 14 dBi gain or less can be used with no cable-loss.

Antenna Type: Flat Panel

RF module was tested and approved with 19 dBi antenna gain with 4.8 dB cable-loss (EIRP Maimum of 14.2 dBm). Any Flat Panel type antenna with 14.2 dBi gain or less can be used with no cable-loss.

XBee ZNet 2.5 RF Module

The following antennas have been tested and approved for use with the embedded XBee ZNet 2.5 RF Module:

- Dipole (2.1 dBi, Omni-directional, Articulated RPSMA, Digi part number A24-HABSM)
- Chip Antenna (-1.5 dBi)
- Attached Monopole Whip (1.5 dBi)

XBee PRO ZNet 2.5 RF Module

The following antennas have been tested and approved for use with the embedded XBee PRO ZNet 2.5 RF Module:

- Dipole (2.1 dBi, Omni-directional, Articulated RPSMA, Digi part number A24-HABSM)
- Chip Antenna (-1.5 dBi)
- Attached Monopole Whip (1.5 dBi).

Canada (IC)

Labeling Requirements

Labeling requirements for Industry Canada are similar to those of the FCC. A clearly visible label on the outside of the final product enclosure must display the following text:

Contains Model XBee ZNet 2.5 Radio, IC: 4214A-XBEE2

The integrator is responsible for its product to comply with IC ICES-003 & FCC Part 15, Sub. B - Unintentional Radiators. ICES-003 is the same as FCC Part 15 Sub. B and Industry Canada accepts FCC test report or CISPR 22 test report for compliance with ICES-003.

If it contains an XBee PRO ZNet 2.5 OEM Module, the clearly visible label on the outside of the final product enclosure must display the following text:

Contains Model XBee PRO ZNet 2.5 Radio, IC: 1846A-XBEEPRO2

The integrator is responsible for its product to comply with IC ICES-003 & FCC Part 15, Sub. B - Unintentional Radiators. ICES-003 is the same as FCC Part 15 Sub. B and Industry Canada accepts FCC test report or CISPR 22 test report for compliance with ICES-003.

Transmitters for Detachable Antennas

This device has been designed to operate with the antennas listed in table A-3 and having a maximum of 17.5 dB. Antennas not included in this list or having a gain greater than 17.5 dB are strictly prohibited for use with this device. The required antenna impedance is 50 Ω

Detachable Antenna

To reduce potential radio interference to other users, the antenna type and gained shuold be so chosen that the equivaleny istropically radiated power (e.i.r.p.) is not more than permitted for successful communication

© 2008 Digi International, Inc.

Appendix C: Migrating from the 802.15.4 Protocol

This following are some of the differences in the ZigBee firmware assuming familiarity with the 802.15.4 application:

- ZigBee Command Set
- Address Assignment
- API / AT Firmware Versions

Also, refer to the "Getting Started" section for more information.

ZigBee Command Set

Modified Commands

- CH - Read Only command that displays the operating channel that was selected from SC.
- MY - Read Only command that displays the assigned 16-bit network address of the device.
- AI - ZigBee definitions added to this command. See documentation.
- A1, A2 and CE commands are not supported.

New Commands

- NJ (Node Join Time) - This value determines how long a Coordinator or Router will allow other devices to join to it. This command is supported on Coordinators & Routers only.
- MP (16-bit Parent Network Address). This value represents the 16-bit parent network address of the module.
- BH (Broadcast Hops). This value sets the maximum number of hops for each broadcast data transmission. Setting this to 0 will use the maximum number of hops.

API / AT Firmware Versions

The 802.15.4 firmware supports the AP command for setting the module into No API (AP=0), API without escaping (AP=1), or API with escaping (AP=2) modes. The first digit in the 802.15.4 firmware versions is a '1'.

The ZigBee firmware comes in different versions to support the API interface (AP 1, 2 modes) or the AT command set (AP 0 mode). The first digit in the ZigBee firmware versions is an '8'.

The following is a list of firmware versions:

- 1.0xx - Coordinator, AT Command support (Transparent Mode)
- 1.1xx - Coordinator, API support (AP 1, 2)
- 1.2xx - Router, End Device, AT Command support (Transparent Mode)
- 1.3xx - Router, End Device, API support (AP 1, 2)

Appendix D: Additional Information

1-Year Warranty

XBee ZNet 2.5 RF Modules from Digi, Inc. (the "Product") are warranted against defects in materials and workmanship under normal use, for a period of 1-year from the date of purchase. In the event of a product failure due to materials or workmanship, Digi will repair or replace the defective product. For warranty service, return the defective product to MaxStream, shipping prepaid, for prompt repair or replacement.

The foregoing sets forth the full extent of MaxStream's warranties regarding the Product. Repair or replacement at MaxStream's option is the exclusive remedy. THIS WARRANTY IS GIVEN IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, AND DIGI SPECIFICALLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL DIGI, ITS SUPPLIERS OR LICENSORS BE LIABLE FOR DAMAGES IN EXCESS OF THE PURCHASE PRICE OF THE PRODUCT, FOR ANY LOSS OF USE, LOSS OF TIME, INCONVENIENCE, COMMERCIAL LOSS, LOST PROFITS OR SAVINGS, OR OTHER INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT, TO THE FULL EXTENT SUCH MAY BE DISCLAIMED BY LAW. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES. THEREFORE, THE FOREGOING EXCLUSIONS MAY NOT APPLY IN ALL CASES. This warranty provides specific legal rights. Other rights which vary from state to state may also apply.

Contact Digi

Free and unlimited technical support is included with every Digi Radio Modem sold. For the best in wireless data solutions and support, please use the following resources:

Technical Support:	Phone.	(866) 765-9885 toll-free U.S.A. & Canada (801) 765-9885 Worldwide
	Live Chat.	www.digi.com
	E-Mail.	support@digi.com

Digi’s office hours are 8:00 am - 5:00 pm [U.S. Central Standard Time]